

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

RÉALISATION D'UN AGENT TUTEUR INTELLIGENT CONSCIENT

MÉMOIRE
PRÉSENTÉ
COMME EXIGENCE PARTIELLE
DE LA MAÎTRISE EN INFORMATIQUE

PAR
MOHAMED GAHA

MAI 2008

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je souhaite remercier avant tout mon directeur de recherche Roger Nkambou qui m'a accueilli au sein de son laboratoire. Il m'a donné la possibilité d'intégrer une équipe dynamique et de travailler sur un projet formidable. Je lui suis très reconnaissant pour l'énorme soutien qu'il m'a accordé. Merci.

Un remerciement spécial pour tous les membres du GDAC, cette formidable famille avec laquelle j'ai passé le plus clair de mon temps. Daniel Dubois, Philippe Fournier-Viger, Patrick Homeyer et Michael Watrelot sont beaucoup plus que de simples collègues, mais des amis et parfois des confidents.

Une pensée très particulière est adressée à mon grand ami Usef Faghihi, formidable ami au cœur d'Or. À Khaled Belghith avec qui j'ai passé des journées mémorables à refaire le monde.

Pour finir, je souhaite exprimer toute ma gratitude à celle qui a su me comprendre et m'aider dans mes moments difficiles, ma compagne Takoua Boukhris.

TABLE DES MATIÈRES

LISTE DES FIGURES.....	v
LISTE DES TABLEAUX	vii
RÉSUMÉ	viii
INTRODUCTION	1
CHAPITRE I : MISE EN CONTEXTE	
1.1 Cadre de recherche	3
1.2 Problématique	5
1.3 Objectifs	6
CHAPITRE II : SYSTÈME TUTEUR INTELLIGENT	
2.1 Modèle de l'apprenant	8
2.2 Le Module du domaine	9
2.3 Module pédagogique.....	11
2.4 Module de Communication	13
2.5 Exemples de STI : les tuteurs cognitifs.....	13
CHAPITRE III : DU RÔLE DE LA CONSCIENCE	
3.1 Définition de la conscience	18
3.2 Systèmes conscients et conscience artificielle	20
3.2.1 Le système conscient GLAIR	21
3.2.2 CLARION.....	23
3.2.3 Le système IDA.....	24
3.3 Théorie de l'atelier global	27
CHAPITRE IV : L'AGENT CTS	
4.1 Architecture du CTS	31
4.2 Modules, traitement et communication.....	35
4.3 Cycle cognitif.....	40

4.4	Forces de CTS	42
4.5	Faiblesses de CTS	46
CHAPITRE V : APPORTS ET MODIFICATIONS DE CTS		
5.1	Communication CTS/Apprenant	54
5.2	Simulation de modèles.....	57
5.3	Enrichissement des scénarios	59
5.4	Les microprocessus de raisonnement.....	63
5.5	Agrégation de l'information.....	65
5.6	Description du Réseau des Actes (RA).	72
5.7	Modification du fonctionnement du Réseau des Actes.....	77
CHAPITRE VI : ÉMOTION LORS DU TUTORAT		
6.1	Le pourquoi de l'émotion (Gaha, Dubois et Nkambou, 2008).....	87
6.2	Émotion et cognition.....	88
6.3	Description théorique	90
6.3.1	Excitateurs externes.....	90
6.3.2	Excitateurs internes	92
6.4	Description fonctionnelle.....	93
6.5	Délibération et remise en cause	96
6.6	Constats et tendances.....	99
CONCLUSION		100
BIBLIOGRAPHIE.....		102

LISTE DES FIGURES

Figure	page
Figure 1.1 : Poste de contrôle de CANADARM2	3
Figure 1.2 : Simulateur du poste de contrôle du Bras.....	5
Figure 2.1 : Architecture d'un <i>STI</i>	8
Figure 2.2 : Modèle <i>overlay</i> dans sa version standard et étendue.....	9
Figure 2.4 : Environnement virtuel de STEVE.....	15
Figure 2.5 : Simulateur de situation TLTS.....	16
Figure 3.1 : Architecture de GLAIR	21
Figure 3.2 : Architecture de CLARION	23
Figure 3.3 : Architecture d'IDA.....	26
Figure 4.1 : Architecture de CTS.....	32
Figure 4.2 : Vue conceptuelle générale de CTS	33
Figure 4.4 : Type de coalition possible	37
Figure 4.5 : Une séquence rudimentaire du Réseau des Actes.....	38
Figure 4.6 : Exemple générique d'un Acte.	39
Figure 4.7 : Organisation des packages de CTS	44
Figure 4.8 : Éditeur du RA.....	ii
Figure 4.9 : Scénarios de délibération.....	48
Figure 5.1 : Exemple de fenêtres informative et interrogative.....	54
Figure 5.2 : Portion du premier scénario du RA.	61
Figure 5.3 : Scénario pour donner un indice	63
Figure 5.4 : Structure du réseau d'information dans la Mémoire de Travail.....	68
Figure 5.5 : Coalition à former	69
Figure 5.6 : Dispersion de l'énergie dans le RA	76

Figure 5.7 : Exemple de liens OU.....	81
Figure 5.8 : Exemple de liens ET.....	82
Figure 6.1 : Les ajouts faits à CTS	86
Figure 6.2 : Pouvoir discriminant des émotions.....	94
Figure 6.3 : Pouvoir discriminant du réseau.....	96
Figure 6.4 : Phase d'évolution d'une délibération	98

LISTE DES TABLEAUX

Tableau	page
Tableau 4.1 : Traduction de l'information	36
Tableau 5.1 : Exécution d'un Acte d'affichage	56
Tableau 5.2 : Mode de communication du Modèle du domaine.....	58
Tableau 5.3 : Mode de communication du Modèle de l'apprenant	59
Tableau 5.4 : Récapitulatif des interactions.....	71
Tableau 5.5 : Récapitulatif des interactions.....	72

RÉSUMÉ

Pour améliorer le rendement des STI, de nombreux investissements matériels et immatériels ont été faits (Starkman, 2007). Cependant, les STI demeurent complexes et leur implémentation coûteuse (Aleven, 2006). En effet, plus l'apprentissage dispensé par le STI est personnalisé, plus le traitement computationnel est complexe.

Réaliser et finaliser un STI capable d'évoluer de manière autonome dans un environnement complexe et riche en informations serait d'un grand intérêt. C'est à cette délicate réflexion que s'attaque mon mémoire.

Tout su long de mon mémoire je présente un agent tuteur cognitif nommé CTS. Il repose sur une architecture semblable à un modèle psychologique de la conscience humaine. CTS cherche à simuler le fonctionnement de la conscience et ainsi tirer profit des phénomènes lui affairant.

L'hypothèse de base est que les mécanismes de la conscience peuvent conférer au STI un comportement lui permettant de mieux gérer la complexité liée à l'environnement afin de prendre les décisions tutorielles de bonne qualité lors d'une séance d'entraînement à l'usage du bras.

Mots clés: Agent cognitif, conscience, Réseau des actes, microprocessus.

INTRODUCTION

Aujourd'hui, la connaissance constitue un levier essentiel pour tout développement économique et social. C'est elle qui autorise la création de la richesse en habilitant les humains à produire mieux et plus efficacement. La connaissance serait au cœur de l'avantage comparatif entre les organisations et entre les pays. Elle est à la base de leur puissance et c'est elle seule qui garantit leur pérennité et leur suprématie. Toutefois, la connaissance n'est point un savoir fini, un bagage cognitif figé. L'immobiliser, c'est la rendre dépassée. La connaissance, pour demeurer efficace, est appelée à se renouveler et à se développer. D'où toute l'importance de l'apprentissage et de la transmission des connaissances.

Les sciences informatiques ont joué un rôle important dans la propagation et dans le partage des connaissances. Les Technologies de l'Information et de la Communication (TIC) ont grandement favorisé la diffusion et le partage des savoirs et des expertises (Dieng, 2000). Depuis la fin des années 1940, des tentatives pour développer des systèmes informatiques pour enseigner ont émergé¹. C'est ainsi que de nouveaux domaines de recherche pour l'enseignement ont vu le jour et connu depuis sous le nom de CBT « *Computer-Based Training* » ou de CAI « *Computer Aided Instruction* ». À présent, les **Systèmes Tutoriels Intelligents (STI)** se présentent comme une évolution significative de ces anciens modèles d'enseignement. Ils sont moins linéaires et dispensent un cours et un encadrement personnalisé selon les préférences et les besoins de l'apprenant.

¹ Le besoin se faisait ressentir quand les premiers soldats revenus de la Deuxième Guerre mondiale ont eu besoin de retourner vers les Universités pour reprendre leurs études (Joseph Beck, 1996).

Pour améliorer le rendement des STI, de nombreux investissements matériels et immatériels ont été faits (Starkman, 2007). Cependant, les STI demeurent complexes et leur implémentation coûteuse (Aleven, 2006). En effet, plus l'apprentissage dispensé par le STI est personnalisé, plus le traitement computationnel est complexe. Cette complexité résulte principalement des nombreux facteurs inhérents à l'apprenant (ses préférences, son profil, ses forces et faiblesses, ses connaissances) et à l'élaboration de cours. Ces nombreux paramètres rendent le système d'apprentissage faiblement déterministe et lui confère un comportement flexible. En effet, plus l'éventail comportemental est large, plus le risque d'erreurs est grand (Davis, 2004).

Réaliser et finaliser un STI capable d'évoluer de manière autonome dans un environnement complexe et riche en informations serait d'un grand intérêt. C'est à cette délicate réflexion que s'attaque mon mémoire.

CHAPITRE I : MISE EN CONTEXTE

1.1 Cadre de recherche

Lors de la construction de la SSI (Station Spatiale Internationale), il a été prévu d'y installer un Bras robotisé capable de déplacer des charges (notamment des astronautes) et d'assembler des parties de la station. Ce Bras robotisé nommé CANADARM2 est un engin complexe et doit être manœuvré avec prudence par les astronautes.

La manipulation du Bras se fait à travers un poste de contrôle (Robotic Work Station) qui comporte entre autres trois écrans reliés à une dizaine de caméras se trouvant sur la SSI (voir Figure 1.1 : Poste de contrôle de CANADARM2). Ces caméras positionnées à des emplacements spécifiques donnent une vue de l'extérieur de la station. Elles constituent les seules sources visuelles dont dispose l'astronaute pour commander le Bras.



Figure 1.1 : Poste de contrôle de CANADARM2

La manipulation du Bras robotisé CANADARM2 est une tâche difficile. Ce dernier dispose de sept joints/degrés de liberté, ce qui le rend difficile à mouvoir. Ainsi, pour faire bouger le *Bras canadien*, l'astronaute doit tenir compte d'une multitude d'aspects tels que la position et configuration initiale du Bras ou l'effet du mouvement d'un joint sur l'ensemble du Bras.

Ainsi, avant chaque manipulation et afin d'éviter tout risque de collision avec la SSI, l'astronaute prend beaucoup de précautions.

L'autre difficulté relative à la manipulation du Bras réside dans le fait que l'astronaute se doit de déterminer de manière optimale les trois caméras à sélectionner sur le poste de contrôle pour obtenir les meilleures vues possibles. Le poste ne permet, à toute fin pratique, d'afficher que trois caméras à la fois et celles-ci sont généralement à changer tout au long de la manipulation du Bras. L'opération de sélection des meilleures vues est difficile. Elle nécessite une bonne connaissance de la Station Internationale ainsi que de la position exacte des caméras. La tâche est d'autant plus ardue que les caméras peuvent être « à l'endroit » ou « à l'envers » sur la Station. L'astronaute se doit de développer ses aptitudes spatiales ce qui requiert beaucoup de temps d'entraînement (Dubois, Nkambou et Homeyer, 2006).

Au laboratoire GDAC (**G**estion, **D**iffusion et **A**cquisition de **C**onnaissances), en partenariat avec l'Université de Sherbrooke et l'Agence spatiale canadienne, un simulateur doté d'un système tuteur a été implémenté, nommé RomanTutor (Kabanza, Nkambou et Belghith 2005; Nkambou, Belghith et Kabanza, 2006). Ce dernier offre une simulation virtuelle du poste de contrôle du CANADARM2 (voir Figure 1.2 : Simulateur du poste de contrôle du Bras.). Le système tuteur permet essentiellement de planifier automatiquement une trajectoire menant de la position/configuration actuelle du Bras à sa destination/configuration finale. Il offre des exercices de manipulation du Bras où il s'agit de mouvoir celui-ci d'une position initiale à une position finale. Toutefois, ce tuteur n'est pas un système tutoriel intelligent complet. En effet, il ne prend pas en considération les particularités individuelles de l'apprenant et ne dispose pas de réactions tutorielles.

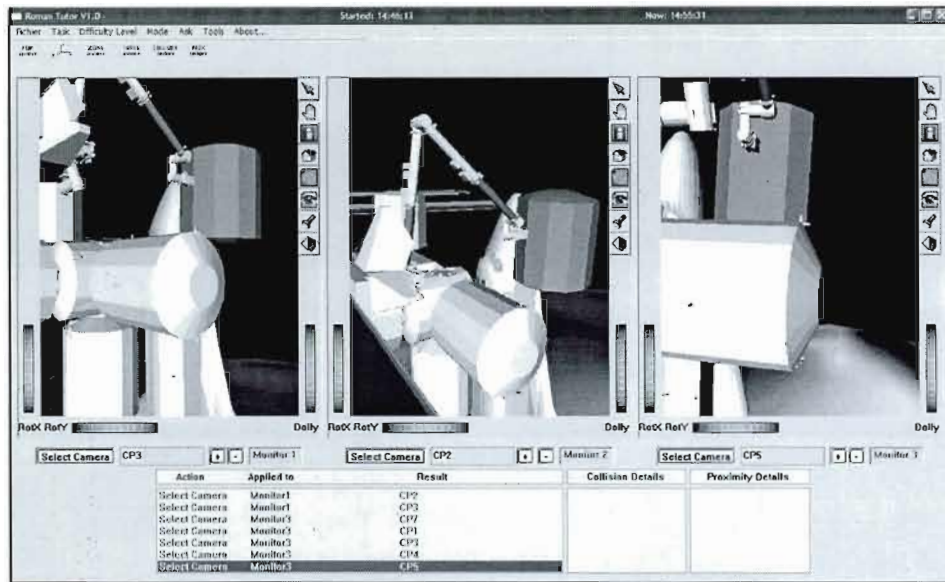


Figure 1.2 : Simulateur du poste de contrôle du Bras.

1.2 Problématique

Dans sa version actuelle, RomanTutor ne permet pas de mettre en place un enseignement adaptatif. Il ne dispose pas suffisamment de profondeur pour offrir des réactions tutorielles de qualité qui tiennent compte du profil de l'utilisateur. Il offre donc un service tutoriel limité qui ne permet pas une articulation profonde sur les erreurs de l'apprenant. Ce type d'articulation nécessite de prendre en compte plusieurs sources d'information disponibles lors d'une session d'apprentissage. Pour ce faire, il y a lieu de se demander :

1. Comment compléter RomanTutor afin de le transformer en un STI capable de dispenser un encadrement de qualité avec un haut niveau de performance?
2. Sur quel modèle d'architecture cognitive peut-on appuyer ce STI afin d'être en mesure d'évoluer de manière autonome dans un environnement changeant et imprévisible?

Ces deux questions forment le cœur de la problématique de mon mémoire. Tout au long de ma rédaction, je vais apporter des éclaircissements sur la manière dont on a procédé au laboratoire GDAC pour répondre à ces attentes.

1.3 Objectifs

Lorsque, j'ai rejoint l'équipe du GDAC en mai 2006, un projet de développement d'un STI qui vise à compléter le système RomanTutor était en cours. Il s'agissait d'un **agent** tuteur intelligent basé sur une architecture dite « *consciente* ». Cet agent est nommé CTS pour 'Conscious Tutoring System'. Le fonctionnement de CTS repose sur une architecture semblable à un modèle psychologique de la conscience humaine. CTS cherchait à simuler le fonctionnement de la conscience et ainsi tirer profit des phénomènes lui affairant.

L'hypothèse de base était que les mécanismes de la conscience pouvaient conférer au STI un comportement lui permettant de mieux gérer la complexité liée à l'environnement afin de prendre les décisions tutorielles de bonne qualité lors d'une séance d'entraînement à l'usage du bras.

Lorsque j'ai repris la plateforme RomanTutor suite aux travaux réalisés par Patrick Hohmeyer dans le cadre de sa maîtrise (Hohmeyer, 2006), mon objectif était de compléter et de finaliser cet agent afin de le rendre fonctionnel. Les travaux de Patrick ont permis de développer un cadre pour l'agent inspiré notamment du code de IDA, une coquille d'agent artificiel conscient reçue de l'Université de Memphis (Franklin, Kelemen et McCauley, 1998). Bien que le noyau de l'agent ait été créé par Patrick, ainsi que certains modules importants (le module de perception par exemple), l'agent ne réalisait encore aucune fonction tutorielle et n'était pas en mesure d'interagir avec l'apprenant. Il ne disposait pas encore de tous les modules et fonctionnalités qui composent un STI.

Mon travail de maîtrise avait donc pour objectif

- 1) de compléter la coquille de l'agent CTS;
- 2) d'y intégrer des réactions tutorielles pertinentes;
- 3) de greffer CTS dans RomanTutor.

1.4 Organisation du mémoire

Afin de répondre à ces objectifs, j'analyserai, dans le deuxième et troisième chapitre, les principaux concepts et approches inhérentes à ma problématique. Je me focaliserai sur les STI et décrirai notamment ses caractéristiques et les modules le composant. Par la suite, j'aborderai les notions de conscience artificielle et en décrirai les nombreux phénomènes. Cette analyse servira comme référence à l'analyse ultérieure.

Dans le chapitre quatre, j'entamerai une description des aspects pratiques de l'agent tuteur développé au GDAC. Je décrirai l'état dans lequel était l'agent quand je l'ai repris au tout début de mon projet. Les détails fonctionnels, les forces et les limites de l'agent y seront abordés. Je reprendrai notamment les principaux problèmes de l'agent. Je les analyserai et j'avancerai les solutions pour les résoudre.

Dans le chapitre cinq, je mets en avant les apports fonctionnels introduits, les extensions et les améliorations réalisées pour finaliser le système et le rendre opérationnel et plus stable. Le chapitre six présente une extension théorique possible du tuteur CTS afin d'y intégrer le concept d'émotion. Enfin, dans la conclusion, je reprendrai l'essentiel de mon apport, je mentionnerai les limites de ma recherche ainsi que les perspectives.

CHAPITRE II : SYSTÈME TUTEUR INTELLIGENT

L'agent CTS est avant tout un Système Tuteur Intelligent et il se doit d'obéir à certaines contraintes de modularité. En effet, en plus d'un module de communication qui sert d'interface entre le système et l'apprenant, un STI se compose d'au moins trois modules essentiels : le Modèle de l'apprenant, le Module du domaine (comprenant essentiellement un Modèle expert et un Modèle des connaissances du domaine) et le Module pédagogique (voir Figure 2.1 : Architecture d'un STI).

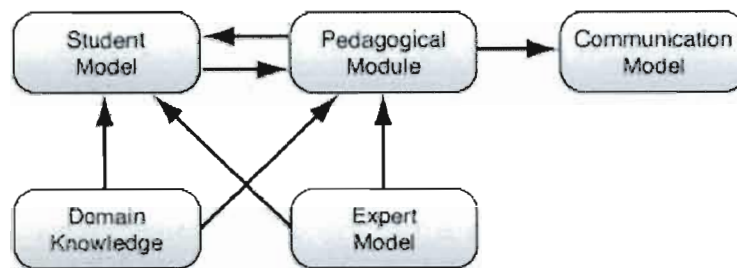


Figure 2.1 : Architecture d'un STI (Beck, 1996)

2.1 Modèle de l'apprenant

Le Modèle de l'apprenant est un élément essentiel aux STI. Il constitue l'élément central qui distingue les STI des systèmes CAI ou CBT qui ne prennent pas en compte les mécanismes de l'apprentissage humain (Urban-Lurain, 1996). Le Modèle de l'apprenant permet un apprentissage personnalisé et ciblé qui tient compte de la progression de l'apprenant lors de la séance d'apprentissage. Il participe à différentes tâches durant la séance d'enseignement et intervient lors de certaines phases de planification du cours. Il peut être sollicité dans le processus de diagnostic lors des erreurs commises par l'apprenant et

permet ainsi d'expliciter les causes probables de ces erreurs (Ragnemalm, 1996). Les informations encodées par ce modèle permettent au Module pédagogique de prendre des décisions tutorielles nécessaires.

Le Modèle de l'apprenant a pour vocation de contenir et de collecter activement toutes les informations relatives à un apprenant (par exemple : le niveau de connaissances et les aptitudes, le niveau de concentration, le style d'interaction préféré par l'apprenant, etc.). Ces informations représentent les croyances du système vis-à-vis de l'état de connaissances de l'apprenant à un instant donné. Ces croyances peuvent être certaines ou probables. Parmi les approches les plus usuelles pour représenter les connaissances de l'apprenant, il existe le Modèle en superposition (*Overlay*) qui considère que les connaissances de l'apprenant sont un sous-ensemble des connaissances de l'expert. En général, le Modèle *overlay* agit de manière à comparer les actions de l'apprenant avec celui de l'expert.

Le Modèle de perturbation (*buggy model*) qui est une évolution du Modèle *overlay* suppose que l'apprenant dispose de connaissances communes avec l'expert ainsi que de connaissances fausses propres à l'apprenant (voir Figure 2.2 : Modèle *overlay*). Ce modèle permet donc de représenter à la fois les bonnes connaissances et les mauvaises connaissances de l'apprenant. Le Modèle de perturbation représente ainsi les erreurs les plus communes qu'un apprenant peut faire lors de la résolution d'un exercice donné. Cette extension permet un meilleur diagnostic des lacunes de l'apprenant et octroie au système la capacité à remédier plus efficacement.

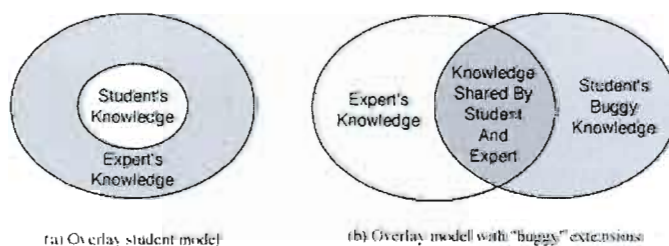


Figure 2.2 : Modèle *overlay* dans sa version standard et étendue.

2.2 Le Module du domaine

Ce module représente le **Modèle des connaissances** et le **Modèle expert** qui forment le cœur du système. Tout comme un enseignant humain, un STI doit préalablement connaître et maîtriser les notions qu'il enseigne.

Dans le Modèle des connaissances, toutes les connaissances du domaine y sont représentées. Il constitue le modèle le plus important, car sans lui il n'y aura rien à faire apprendre (Beck, 1996). C'est à partir de ce module du domaine qu'il y a construction des éléments qui forment le cours. Des métadonnées peuvent être rajoutées afin de permettre au système de choisir la ressource la plus judicieuse à présenter.

Selon la manière dont est représentée la connaissance (formalisme utilisé, niveau de granularité, expressivité du formalisme) dépendra en grande partie l'efficacité du système. La représentation doit se faire de manière à ce qu'il soit possible d'enregistrer et de manipuler la connaissance par le STI. En effet, les connaissances du domaine doivent être encodées de manière suffisamment fine pour que le STI soit capable de faire de l'inférence et de simuler le comportement d'un expert humain (Fournier-Viger, 2006).

Des recherches avancent qu'il faut parcelliser les connaissances pour les représenter de manière plus fine qu'une traditionnelle représentation en déclaratif/procédural. Les travaux de R. Gagné (Zinkiewicz, Hammond et Trapp, 2003) plaident pour une représentation à cinq catégories de connaissances. Les systèmes de représentation MOT (**M**odeling using **O**bject **T**ypes) et CREAM (**C**urriculum **R**Epresentation and **A**cquisition **M**odel) (Nkambou, Frasson et Gauthier, 2003) se basent sur une telle distinction.

En complément au Modèle des connaissances, le Modèle expert est celui qui est capable de réaliser des inférences et de simuler le comportement d'un expert humain en s'appuyant sur les connaissances modélisées. L'expert est celui qui détient le *savoir-faire* du domaine. Il est capable de résoudre un problème dans le domaine et constitue la ressource à qui s'adresse le STI pour résoudre un problème donné. Le Modèle expert permet de montrer les étapes à suivre pour la résolution d'un problème. Je cite à titre d'exemple le module

planificateur (Belghith *et al.*, 2006) qui forment le Modèle expert de RomanTutor. Il est capable de résoudre des problèmes de manipulations du bras Canadien.

Un STI est en mesure de déterminer si l'apprenant est sur la bonne voie grâce au Modèle expert. Si, par contre, l'apprenant échoue dans la résolution de l'exercice, le système est capable de déterminer l'étape où l'apprenant a eu de la difficulté et de présenter la prochaine étape à faire sans pour autant donner la solution.

Il arrive que dans le Modèle des connaissances, les exercices et les solutions soient décrits tacitement dans le Modèle des connaissances. Dans d'autres STI, c'est au système de générer les exercices ainsi que leurs solutions. Toutefois, développer à la volée (*just in time*) les exercices est une approche bien plus compliquée et coûteuse à implémenter que le simple fait de présenter des séquences d'exercices préenregistrées (Ong et Ramachandran, 2003). Cependant, le fait que les exercices soient générés octroie au système un plus grand choix d'exercices et donc la capacité de mieux cibler les erreurs de l'apprenant.

2.3 Module pédagogique

Ce module encode la connaissance nécessaire pour la mise en œuvre de stratégies pédagogiques. Il contient des séquences d'Actes pédagogiques à mettre en place lors de la séance d'apprentissage. C'est lui qui décide quel contenu présenter à l'apprenant. Le Modèle de l'apprenant et le Module du domaine interviennent grandement lors du processus de construction du cours et fournissent les données en entrées au Module pédagogique.

Le Module pédagogique agit en sélectionnant une stratégie pédagogique pour l'enseignement du domaine. Par exemple, le système peut décider d'utiliser une méthode socratique ou en spirale. Dans le cas où le STI dispose de plusieurs stratégies pédagogiques, le Module pédagogique s'appuie sur le Modèle de l'apprenant et le Modèle des connaissances afin de sélectionner la meilleure stratégie à mettre en œuvre. Il y a lieu pour

le système de déterminer les détails de chaque action et du contenu à présenter. Le Module pédagogique peut décider de prendre trois actions possibles :

1. La sélection de concept : pour sélectionner un nouveau concept, le modèle doit examiner le Modèle de l'apprenant afin de déterminer le concept qui correspond le mieux aux besoins de l'apprenant. Une fois le concept sélectionné, le Modèle des connaissances offre le contenu à présenter.
2. La génération de problèmes : après la sélection d'un concept, il y a lieu de générer un problème à présenter à l'apprenant. Le niveau de granularité du problème est déterminé par le Modèle des connaissances en fonction du niveau d'habileté de l'apprenant.
3. Fournir un compte-rendu : quand l'apprenant a besoin d'aide, le système doit être en mesure de fournir un compte-rendu ou de l'aide afin de l'orienter dans la résolution de l'exercice ou du problème. Toute la difficulté réside dans la détermination du niveau de granularité à fournir. Peu ou trop d'aide mène à la frustration de l'apprenant et le dosage doit être savamment ajusté. Le Modèle de l'apprenant va jouer le rôle de juge pour déterminer le niveau de détail de l'aide dépendamment du niveau d'expertise de l'apprenant. Le Modèle des connaissances offrira le contenu de l'aide.

Le Module pédagogique interfère avec les 3 autres modules composant le STI et joue le rôle de chef d'orchestre pour coordonner le fonctionnement de tous les modules. En effet, les interactions entre les différents modules sont nombreuses. Il en découle un système complexe à concevoir et à implémenter.

De nos jours, les développeurs s'orientent de plus en plus vers des **systèmes auteurs** tel que CTAT (**Cognitive Tutor Authoring Tools**) (McLaren *et al.*, 2004). Ils offrent une suite d'outils de développement pour la réalisation de tuteur informatique. Dans le cas de CTAT, on

permet aux non-informaticiens de développer un pseudo-tuteur à travers une programmation par démonstration sans avoir recours à une programmation poussée.

2.4 Module de Communication

Le Module de communication est celle qui interagit avec l'utilisateur. Il affiche les écrans et les dialogues tout au long de la séance d'apprentissage. Il constitue le contact entre l'homme et la machine. Selon la manière dont est affichée l'information sur l'écran, il peut avoir un impact chez l'apprenant.

Le Module de Communication est aussi responsable de détecter les actions de l'utilisateur. Il agit comme un régulateur entre les flux de communication entrants et sortants du système. Ce module va agir comme traducteur en transformant les représentations internes de la machine en une interface visuelle compréhensible par l'apprenant.

L'importance de ce module va dépendre grandement du type d'apprentissage à mettre en place. Dans le cas où l'apprentissage concerne des connaissances procédurales, où l'environnement de travail est une reproduction virtuelle de la réalité, l'interface de communication va prendre toute son importance.

Urban-Lurain (1996) donne deux autres rôles au Module de communication. Il soutient que, lors de la résolution d'un problème, les apprenants frustrés fournissent des réponses contradictoires ou incomplètes. Comme premier rôle, l'interface doit être capable de résoudre l'ambiguïté d'une réponse fournie par l'apprenant. Comme deuxième rôle, l'interface doit être capable de guider l'apprenant à travers le dédale d'informations qui lui est présenté. En effet, il arrive que l'étudiant passe outre des détails importants d'une leçon.

2.5 Exemples de STI : les tuteurs cognitifs

Les premiers systèmes d'enseignement qui ont fait preuve d'efficacité datent de l'année d'apparition de la théorie ACT* « the **A**daptive **C**ontrol of **T**hought » d'Anderson (Anderson, 1983). Celle-ci décrit une approche cognitive de l'esprit humain transposable dans divers champs d'applications informatiques, notamment dans les STI. La théorie d'Anderson a permis la construction de STI relativement performants (LISP Tutor: Anderson, Conrad et Corbett, 1989; Geometry Tutor : Anderson, Boyle et Yost, 1985). De nos jours on parle davantage de ACT-R « the **A**daptive **C**ontrol of **T**hought - **R**elational » (Anderson, 1993) qui se veut un raffinement du mode opératoire d'ACT*.

Le système PAT (**P**actical **A**lgebra **T**utor) s'appuie sur la théorie cognitive d'ACT-R. Ce STI enseigne des concepts d'algèbre. Les cours dispensés par PAT sont répartis en plusieurs sections. Chaque section fait appel à des habiletés différentes. Pour que l'apprenant réussisse une section, il doit en maîtriser toutes les habiletés.

PAT utilise un ensemble d'instructions faisant référence à des connaissances déclaratives et procédurales. Le tuteur dirige les étudiants dans la résolution du problème et arrive à diagnostiquer ce que l'apprenant sait et ce qu'il ne sait pas. En diagnostiquant les actions de l'apprenant et en réalisant une correspondance avec les instructions représentées en interne, le système peut fournir de l'aide. Dans PAT, le Modèle de l'apprenant s'appuie sur un Modèle Overlay de type Buggy. Le Modèle des connaissances contient l'ensemble du contenu du cours (Ritter *et al.*, 1998).

Plusieurs études ont démontré que PAT permet d'améliorer les capacités d'assimilation des apprenants. Le succès de PAT tient du fait qu'il utilise une approche cognitive de l'esprit humain (Anderson et Schunn, 2000).

Un autre STI qui intègre un enseignant virtuel doté de capacités cognitives est STEVE (**S**oar **T**raining **E**xpert for **V**irtual **E**nvironments) (Elliott *et al.*, 1999). C'est un agent virtuel pédagogique émotif qui évolue dans un monde virtuel semblable à l'environnement de travail de l'apprenant. L'agent a pour principal objectif d'aider l'apprenant dans la

manipulation et la réparation d'une pompe hydraulique sur un navire de guerre (voir Figure 2.3 : Environnement virtuel de STEVE).

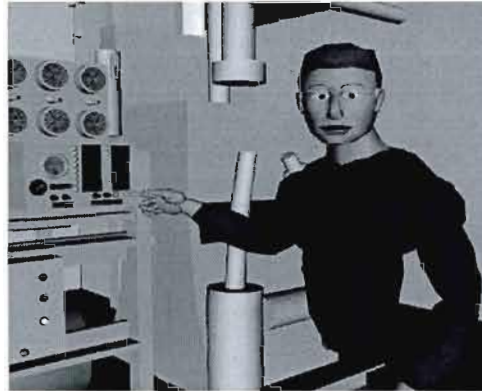


Figure 2.3 : Environnement virtuel de STEVE

Le système cherche à inculquer des connaissances procédurales à travers la manipulation de la pompe. L'immersion de l'apprenant dans l'environnement virtuel donne une meilleure portée à l'apprentissage et rend les interactions plus riches entre apprenant et agent tuteur.

L'agent STEVE communique avec l'apprenant de manière verbale et non verbale, et ce, à travers des gestes, des postures et des expressions faciales. Il dispose de 26 types d'émotions et 70 expressions faciales. Il est appelé à instruire, guider et collaborer avec l'apprenant à travers des opérations de démonstration de tâches, de réponse aux questions de l'apprenant et de fournir de l'aide pour résoudre un problème donné.

STEVE garde dans une mémoire épisodique les situations dans lesquelles il a réalisé ses actions. Pour une démonstration donnée, STEVE est capable d'expliquer à l'apprenant le pourquoi de ses Actes et leurs conséquences sur le fonctionnement de la pompe.

Le Module du domaine de STEVE représente les connaissances procédurales à l'aide d'un réseau procédural (Rickel et Johnson, 1997). Un planificateur d'ordre partiel (« partial-order planning ») est utilisé pour permettre à STEVE de prendre les actions qui s'imposent lors de la manipulation de la pompe. Toutefois, STEVE ne dispose pas de Modèle de l'apprenant et ne réagit que lorsque l'utilisateur lui demande de l'aide.

Un autre Système Tutoriel Intelligent tout récemment développé et qui suscite beaucoup d'intérêt est le Tactical Language Training System (TLTS) (voir Figure 2.4 : Simulateur de situation TLTS). Il a pour finalité d'aider à l'acquisition des aptitudes communicatives élémentaires d'une langue étrangère (Johnson, 2006). Évoluant dans un village virtuel, l'apprenant est appelé à parler la langue étrangère afin de réaliser des missions précises. Il est accompagné tout au long de son apprentissage par un agent d'aide qui l'oriente et le guide.

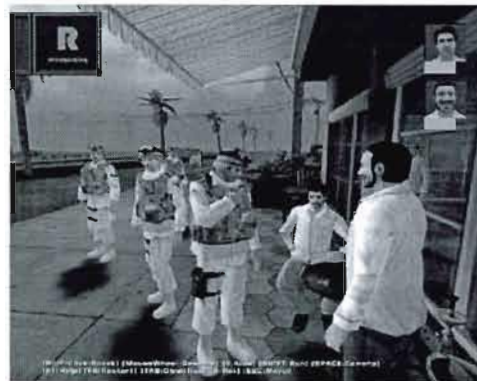


Figure 2.4 : Simulateur de situation TLTS

Le Modèle de l'apprenant arrive à gérer diverses sources de données provenant de l'environnement d'apprentissage. Lors de l'identification de l'utilisateur, son profil est chargé dans le STI ainsi que les informations relatives à sa progression. Le Module

pédagogique surveille les performances de l'apprenant et décide du cheminement du cours et de l'action pédagogique à prendre. Les décisions prises par le Module pédagogique s'appuient essentiellement sur les conclusions extraites du Modèle de l'apprenant. Les données relatives au cours sont stockées dans une base de données sous forme de code XML. Ces données concernent les missions, les leçons, les exercices et des données multimédias. (Johnson *et al.*, 2006).

Un tel système est utilisé par les militaires américains et les résultats obtenus paraissent probants. Les soldats reconnaissent qu'avec ce STI, ils ont appris beaucoup plus en une heure que pendant toute la durée de leur mission à l'étranger (Johnson, 2007).

Au regard de ces systèmes performants, de nombreuses similitudes peuvent être dégagées. En premier lieu, un environnement de travail réaliste, dans lequel évolue l'apprenant, permet une meilleure assimilation des connaissances. Deuxièmement, tenir compte de l'aspect cognitif de l'apprenant optimise ses performances lors du processus d'enseignement. En effet, il est essentiel pour le STI de connaître les aptitudes de l'apprenant afin d'orienter la progression du cours et de mieux cibler ses difficultés. Finalement, un système tuteur qui s'approche du modèle cognitif humain serait, en effet, le plus accompli, car en définitive, le système cherche principalement à reproduire le comportement d'un enseignant dispensant réellement un cours. C'est vers cet objectif que tend notre agent tutoriel CTS.

CHAPITRE III : DU RÔLE DE LA CONSCIENCE

La caractéristique fondamentale de CTS provient principalement de la théorie de l'atelier global (Global Workspace theory) proposée par Baars (1988, 1997). Cette théorie psycho - neurologique propose un cadre général décrivant l'esprit humain et le rôle essentiel que jouent l'attention et la **conscience**. Afin de comprendre l'architecture de CTS et son fonctionnement, je rappelle brièvement ce qu'est la conscience. Volontairement, je ferai abstraction des polémiques relatives à l'existence et aux manifestations de la conscience abordées entre autre par Denett, McGinn et Turing (Blackmore, 2003). Ces interrogations sont, certes, importantes, mais dépassent le cadre et la portée du présent travail. Mon but premier est de faire ressortir l'utilité de la conscience dans le développement d'agents informatiques. Je m'intéresse uniquement aux différents phénomènes liés à la conscience pour le traitement de l'information et la prise d'action.

3.1 Définition de la conscience

La conscience est un phénomène difficile à caractériser et à comprendre. Selon Kirilyuk (2003), il n'existe pas une définition universelle de ce concept et les débats autour de son identité risquent de durer encore longtemps.

La question de savoir ce qu'est la conscience remonte très loin dans l'histoire jusqu'à l'époque de Platon (Koch, 2004). Platon la décrit comme étant notre âme, cette entité immortelle qui se trouve dans un corps mortel. Descartes (Koch, 2004), affirme qu'il existe dans le monde deux types d'êtres. Ceux qui ne sont pas capables de réfléchir tels les animaux et ceux dotés de capacité de réflexion tels les humains. Cette capacité à penser, propre à l'homme, est, ajoute-t-il, ce qui donne lieu à la conscience. Descartes est parmi les premiers à faire de la conscience une caractéristique propre à l'être humain. Baars (1997) et

Sloman & Chrisley (2003) rajoutent que la conscience est l'aboutissement de l'évolution naturelle. Elle est cette entité (ou du moins phénomène) qui joue un rôle prépondérant dans l'adaptation pour faire face aux différents changements de tous les jours. Baars ajoute que la conscience octroie à l'homme une surprenante capacité à traiter diverses sources d'informations sans être saturé et d'évaluer virtuellement des possibilités sans devoir subir les risques associés aux expérimentations physiques (Baars, 1997) (Sloman et Chrisley, 2003).

Chalmers (1995) affirme qu'il n'existe pas un seul problème lié à la conscience, mais plusieurs. Il distingue ainsi une famille de problèmes *faciles* et une autre famille *difficile* à résoudre. Pour Chalmers, la conscience intervient dans les phénomènes et les traitements cognitifs suivants :

1. Habilité à la discrimination, catégorisation et réaction aux stimuli environnementaux.
2. Intégration des informations par un système cognitif.
3. Expression de la situation d'un état.
4. Habilité du système à accéder à ses états internes.
5. Contrôle délibéré de soi.
6. Différenciation entre l'éveil et le sommeil.

Baars, quant à lui, apporte des descriptions plus fines des rôles joués par la conscience. Il décrit dix-huit phénomènes qui lui sont associés. Cette liste inclut des phénomènes tels que la capacité d'accéder à de vastes ressources inconscientes, de prioriser les informations et d'en traiter les plus urgentes, d'enrichir ou de compléter l'interprétation des stimuli, de résoudre des problèmes et de délibérer sur une solution éventuelle, de planifier et de créer des cheminements d'actions pour atteindre des objectifs, de superviser et de contrôler des

actions, d'optimiser les fonctions automatisables et finalement de créer et de maintenir le contexte interprétatif.

Tous ces phénomènes sont d'une grande utilité pour les systèmes informatiques. Prioriser l'information, délibérer pour trouver une meilleure solution et réagir dans un environnement changeant sont autant de voies possibles que les systèmes informatiques cherchent à investiguer et à explorer. Ces préoccupations s'avèrent d'une importance centrale et trouvent des débuts de réponses dans le domaine de l'IA (Intelligence Artificielle). Une des manières de reproduire ces phénomènes est de s'attaquer au cœur du problème en cherchant à reproduire une conscience artificielle. La machine serait ainsi capable de reproduire l'ensemble ou une partie de ces phénomènes.

Imiter le fonctionnement de la conscience et le transposer numériquement dans un agent informatique tel un STI est un avantage certain. Soyons clairs, mon but n'est pas d'octroyer une *quelconque conscience* à la machine, mais d'essayer de tirer profit des avantages de celle-ci en reproduisant certains des phénomènes précédemment cités. En effet, la conscience octroie des possibilités avancées, nécessaires au bon fonctionnement de l'agent tuteur intelligent qui se nomme désormais CTS pour (Conscious Tutoring System).

3.2 Systèmes conscients et conscience artificielle.

De par sa nature, la conscience a longtemps fasciné les hommes et a donné lieu à plusieurs débats et recherches. Durant les dernières décennies, les sciences informatiques se sont intéressées à la conscience et à ses possibilités dans les agents informatiques. Plusieurs systèmes dis « *conscients* » ont ainsi vu le jour (Sun, 1997). Ces systèmes s'appuient sur différentes recherches faites en neurologie, psychologie, psychiatrie, etc. pour reproduire le fonctionnement cérébral humain à travers un programme informatique. En reproduisant le mode de fonctionnement de la conscience, ces systèmes seraient ainsi en mesure de reproduire certains des phénomènes associés à la conscience.

3.2.1 Le système conscient GLAIR

Parmi les tout premiers systèmes qui utilisent une conscience artificielle, c'est-à-dire dans le présent cas, une conscience simulée informatiquement, il y a le système GLAIR (**G**rounded **L**ayered **A**rchitecture with **I**ntegrated **R**easoning). Développé par l'équipe de Hexmoor (Hexmoor, Lammens et Shapiro, 1993), il se compose de 3 couches différentes : la couche connaissance « *Knowledge level* » ; la couche motrice et perceptuelle « *Perceptuo-Motor Level* » et la couche convertisseur sensoriel « *Sensori-Actuator Level* » (Hexmoor, 1995). La partie dite « *consciente* » représente le niveau le plus haut du système qu'est celui des connaissances. Les deux autres niveaux plus bas sont considérés comme inconscients (voir Figure 3.1 : Architecture de GLAIR).

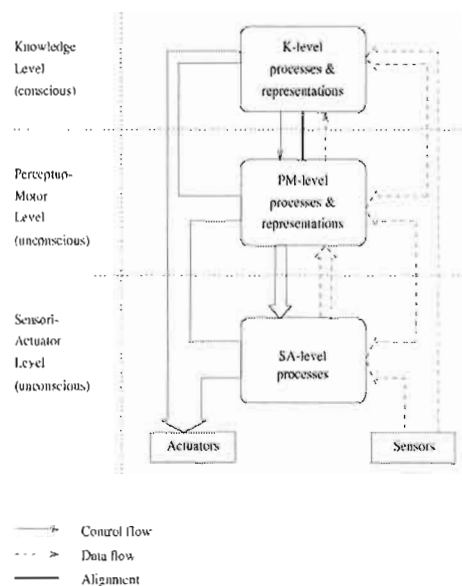


Figure 3.1 : Architecture de GLAIR (Hexmoor, 1995)

Les concepteurs font la distinction entre conscience et inconscience dans leur système en s'appuyant sur des recherches philosophiques. L'Homme réalise une action consciente dans le cas où elle nécessite un traitement relativement lourd et qui fait appel à différentes

sources d'informations, par opposition une action est inconsciente si elle est rapide et ne requiert que très peu de ressources (Crick et Koch, 2003). Ce genre de distinction entre action réfléchie (consciente) et action rapide (inconsciente) se retrouve dans le mode de fonctionnement du système GLAIR.

L'équipe de Hexmoor considère que leur système est conscient, car il est capable (Hexmoor, Lammens et Shapiro, 1993):

1. d'avoir une représentation interne de l'environnement
2. de raisonner à propos de l'environnement
3. de communiquer avec un agent externe à propos de l'environnement.

Le système GLAIR se définit comme un agent autonome, capable de percevoir et d'agir dans un environnement dynamique. Il dispose de capacité d'apprentissage, de raisonnement, de planification et de délibération ce qui le rend très polyvalent. Il a été appliqué à divers types de problèmes² et pour différentes tâches.

D'après Hexmoor (1995), le système marche bien et fournit un comportement intéressant. Toutefois, le système comporte une limite en ce qui concerne l'assise scientifique de l'architecture consciente. Même si cette architecture semble à priori plausible, le bien-fondé scientifique de l'approche consciente adopté pour GLAIR reste sujet à rectification, car ils soulignent n'avoir encore entrepris aucune étude expérimentale visant à vérifier la validité psychologique de leur architecture (Hexmoor, Lammens, Caicedo, et Shapiro, 1993).

Les propriétés qu'ils indiquent comme étant caractéristiques de la conscience semblent être les propriétés de base que l'on retrouve dans des agents dits cognitifs qui n'intègrent pas forcément des phénomènes liés à la conscience tels que définis plus haut.

² Dans l'article d'Hexmoor (Hexmoor *et al.*, 1993), l'architecture de GLAIR est utilisée pour manipuler un simulateur d'avion pour un jeu et pour contrôler un robot serveur.

3.2.2 CLARION

Une autre architecture dite consciente est celle de CLARION (**C**onnectionist **L**earning with **A**daptative **R**ule Induction **ON**-line) développé par Sun (1997). Elle dispose de deux niveaux de représentation dont l'un est conscient et l'autre inconscient. Ces deux niveaux traitent consécutivement les connaissances procédurales et déclaratives (voir Figure 3.2 : Architecture de CLARION). Un troisième module additionnel a été ajouté. Il représente la mémoire épisodique qui renferme les expériences récentes. Les concepteurs de CLARION se sont appuyés sur plusieurs études psychologiques qui font état de la dichotomie entre les connaissances déclaratives et procédurales ainsi que leur mode d'assimilation d'où, d'ailleurs, leur idée d'utiliser une architecture à deux niveaux.

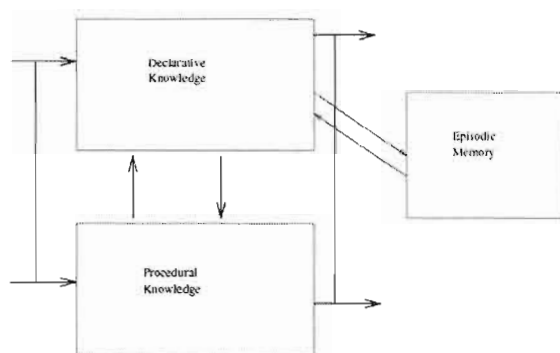


Figure 3.2 : Architecture de CLARION

L'idée de créer CLARION en deux réside dans le fait que l'homme, lors de son apprentissage, fait intervenir en totalité ou en partie des modules inconscients. CLARION s'est inspirée du processus d'apprentissage humain d'où l'utilisation de ce genre d'architecture à deux niveaux. Tout comme GLAIR, CLARION utilise le niveau conscient pour des relations complexes et le niveau inconscient pour des relations simples. L'exécution d'actions dans CLARION est généralement initiée dans la zone inconsciente, mais aboutit à l'activation de la

partie consciente. À son tour la zone consciente affecte le niveau inconscient de manière à apporter des rectifications.

Ainsi, une communication bidirectionnelle s'établit entre les deux niveaux. Les concepteurs de CLARION pensent que le fait de séparer ces deux niveaux permet de mettre en place une synergie et que les deux niveaux sont finalement complémentaires. Cette interinfluence et collaboration ont pour conséquence d'augmenter les performances de CLARION dans l'acquisition de nouvelles connaissances. Ce système est dit *conscient*, car il est capable de simuler certains aspects de la conscience (Sun, 1997):

1. Il a la capacité d'agir et de réagir dans son environnement.
2. Il a la capacité de définir ses propres concepts à propos de l'environnement qui l'entoure.
3. Il a la capacité d'apprendre de nouvelles tâches complexes.
4. Il a la capacité de planifier des actions et des plans avant d'agir.

CLARION utilise certains phénomènes liés à la conscience pour offrir des fonctionnalités riches qui le rendent *intelligent*. Ce genre d'architecture a prouvé son efficacité dans certaines tâches telles que la simulation virtuelle de la navigation à travers un champ miné.

Lorsqu'on compare l'architecture de CLARION avec certains modèles de la conscience tel que celui de Baars (Baars, 1988; 1997), des similitudes ressortent.

3.2.3 Le système IDA

À la fin de l'année 1998, l'équipe de Stan Franklin de l'université de Memphis développe un agent intelligent dit conscient (Franklin, Kelemen, et McCauley, 1998). L'agent se nomme IDA pour (Intelligent Distribution Agent) et se distingue des précédents systèmes conscients par le fait qu'il s'appuie sur un modèle rigoureux et scientifique de la conscience décrit dans les travaux de Baars.

Franklin avance qu'IDA simule les fonctions de haut niveau du cerveau humain. Ainsi, l'agent est capable de reproduire certains traits liés à l'Homme tels que la perception, certaines formes de mémoire, la simulation des émotions, l'exécution d'actions, la génération d'un langage pour la communication et la métacognition (Franklin, 1995).

Une autre particularité de l'agent IDA est qu'il imite à un haut niveau le mode de fonctionnement cérébral humain. Tout comme le cerveau humain dont le traitement est fortement distribué, le fonctionnement d'IDA repose sur ce même principe. Il utilise des *Codelet* (que nous appelons en français *microprocessus*) afin de réaliser des traitements distribués, élémentaires et spécialisés. Les microprocessus correspondent aux **processeurs** dans la théorie de Baars. Ces derniers sont des mini-agents³ capables de réaliser un traitement élémentaire. Ils sont à l'affût d'un élément d'information ou d'une situation particulière qui va les pousser à réagir (Franklin, 1995). Ces microprocessus font généralement tout le travail dans le système IDA.

Tout le fonctionnement d'IDA tourne autour de la Mémoire de travail. Comme précisé précédemment, la majorité du traitement cérébral se fait en parallèle et inconsciemment (voir Figure 3.3 : Architecture d'IDA). Le résultat de ce traitement se retrouve dans la zone tampon que forme la Mémoire de travail. Dans cette dernière d'autres informations peuvent déjà y figurer; elles proviennent des traitements antérieurs ou de la Perception. Dépendamment du contexte, l'information la plus importante et la plus prioritaire est extraite de cette mémoire. L'information sélectionnée devient accessible et connue par tout le reste du système et par conséquent elle est dite *consciente*. Cette information est donc distribuée à l'ensemble des modules et microprocessus de l'architecture pour être traitée. Le cycle se poursuit ainsi indéfiniment et oscille entre conscience et inconscience (Franklin, 2000).

³ Ce sont des agents car ils sont capables de percevoir, de traiter l'information puis d'agir conséquemment à la poursuite de ses buts propres.

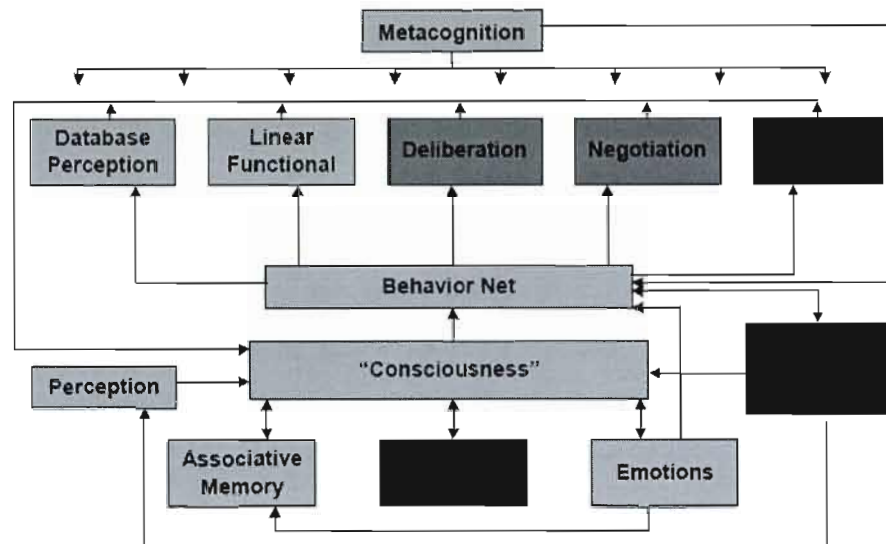


Figure 3.3 : Architecture d'IDA

L'agent IDA est opérationnel et est utilisé dans la marine américaine. Il a pour tâche de négocier avec les marins de l'US NAVY afin de les affecter aux postes disponibles qui leur conviennent le mieux. IDA est capable de communiquer avec les marins en langage naturel *via* courriels interposés. Dans le cas où le marinier n'est pas satisfait de l'affectation proposée par le système, l'agent est en mesure d'entamer une négociation afin de trouver une entente. L'affectation se fait en respectant les contraintes de l'US NAVY (exemple : minimiser les couts de déplacements des marins) et des préférences du marin. L'équipe de Franklin a démontré à travers son agent IDA l'importance du rôle de la conscience dans les traitements informatiques et les intérêts de celle-ci.

Au regard du potentiel d'IDA, il a été décidé au GDAC de développer un système intelligent qui s'appuie sur les concepts de Baars et de l'architecture d'IDA. Afin de saisir en détail le fonctionnement de l'agent CTS et pour en connaître ses forces et faiblesses, il y a lieu de décrire en détail ce qu'est la théorie de l'atelier global de Baars, concept fondamental dans l'architecture d'IDA et par conséquent CTS.

3.3 Théorie de l'atelier global

La « Global Workspace Theory (GWT) » ou la Théorie de l'Atelier Global, de Baars, est une théorie qui modélise le fonctionnement de l'esprit humain avec un rôle explicite pour la conscience. La théorie de Baars présente l'intérêt d'être la première proposition qui expose un cadre complet expliquant le fonctionnement de l'esprit humain. Baars s'appuie sur une foule de données et d'études pour asseoir sa théorie. Il avance que son modèle est une théorie scientifique rigoureuse de la conscience (1997).

Sa théorie se base sur le concept de Mémoire de travail qui se définit comme l'ensemble des structures et des processus utilisés de manière temporaire pour la **manipulation** et la **sauvegarde** de l'information⁴. Ce n'est pas une structure passive, bien au contraire elle peut effectuer certains traitements. La Mémoire de travail est de taille réduite et contient des informations périssables dans le temps (Miller, 1956). Elle peut être vue comme une zone de transition temporaire. Ces constats émanent d'expérimentations et de cas cliniques qui appuient l'existence d'une telle structure dans le cerveau humain.

La Mémoire de travail opère des manipulations sur l'information qui y est existante et joue un rôle actif sur celle-ci. Elle est responsable notamment de l'identification de l'information la plus pertinente à traiter en priorité. Un autre traitement que fait la Mémoire de travail serait aussi la suppression d'informations non pertinentes ou celles qui ne sont plus contextuellement importantes ou significatives (Baddeley et Hitch, 1974).

Baars avance que le traitement dans le cerveau humain se fait de manière répartie, et ce, par des collections de **processeurs** neuronaux efficaces et élémentaires. Le fait que le traitement soit réparti et se passe à différents endroits du cerveau rend le traitement inconscient pour la personne. Ces processus inconscients traitent l'information pour lesquels ils sont spécialisés. L'information traitée est déposée dans la Mémoire de travail. À ce

⁴ D'un point de vue biologique l'information peut être décrite comme formée par un groupe neuronal.

moment précis, si ce résultat prime sur les autres informations faisant partie actuellement de la Mémoire de travail, la personne devient consciente du résultat du traitement. Une fois que l'information est consciente, elle est possiblement retraitée par des microprocessus inconscient et ainsi de suite.

Pour illustrer un tel fonctionnement, je présente ci-après un exemple concret. Supposons qu'une personne vise à résoudre mentalement une opération mathématique simple. L'information représentant l'équation est perçue, traduite en information et déposée dans la Mémoire de travail. Cette information se retrouve avec d'autres informations provenant d'autres sources telles notre mémoire sémantique. Toutes les informations collectées dans la Mémoire de travail vont compétitionner pour accéder à la conscience. Dans le cas où l'information représentant l'équation est jugée la plus pertinente par un arbitre d'attention (Crick et Koch, 2003) elle devient consciente et donc diffusée à tous les autres modules inconscients qui en prennent connaissance. Ces modules qui composent notre cerveau et auxquels nous n'avons pas directement accès réagissent instantanément à cette information. Dans le cas où ces modules reconnaissent l'information et savent la traiter, ils y apportent une partie de la réponse. Dans le cas contraire, ces modules ne réagissent pas. L'information traitée est remise dans la Mémoire de travail et constitue un nouveau pas dans la résolution de l'équation.

La théorie de Baars peut être perçue comme une théâtralisation. Baars compare la Mémoire de travail à la scène qui, selon lui, représente l'élément le plus important du théâtre. La scène, de capacité spatiale limitée, contient les acteurs qui matérialisent les informations présentes dans la Mémoire de travail. La scène n'est pas totalement éclairée et une grande partie baigne dans l'ombre. Les acteurs qui s'y trouvent sont cachés sont invisibles et sont dits inconscients. Sur scène, un projecteur commandé par l'arbitre d'attention éclaire un ou plusieurs acteurs (s'ils forment un groupe) de la scène, ce qui les rend visibles par tous les spectateurs du Théâtre et sont alors dits conscients. Ces acteurs sont mis en lumière s'ils possèdent le message le plus important à faire passer à l'audience par rapport au contexte

actuel. Dans la Mémoire de travail, les acteurs visibles (conscients) et invisibles (inconscients) sont en compétition afin de garder ou d'attirer sur eux le projecteur.

En plus de la scène, il existe ce que Baars appelle *les opérateurs de contexte*. Selon Baars, ces opérateurs tiennent trois rôles principaux : (1) le directeur du théâtre, (2) le personnel et (3) le contrôleur projecteur. Ils supervisent le jeu des acteurs et veillent au bon fonctionnement de l'ensemble. Ces opérateurs surveillent la scène et peuvent influencer son contenu et la direction à donner à la pièce. Pour ce faire, le directeur du Théâtre connaît les buts actuels du système et guide la Mémoire de travail vers une direction particulière.

Dans le théâtre, les spectateurs se trouvent dans l'ombre face à la scène. Ils sont en quelque sorte des acteurs qui observent et réagissent à ce qui se passe sur scène. Ils ne voient que les acteurs éclairés par le spot lumineux. Ils ont la possibilité d'influencer la suite du spectacle. Dans le cas où certains membres de l'audience se sentent capables de contribuer au spectacle ou que l'Acteur sous le projecteur lumineux les interpelle, ils entrent en scène pour devenir possiblement éclairés et donc conscients.

Les spectateurs représentent le traitement inconscient qui se fait dans le cerveau humain. La majorité du travail cérébral se fait, en effet, de manière inconsciente de manière répartie et en parallèle (Dehaene et Naccache, 2001). Le résultat de ce travail caché et inconscient peut se retrouver sur la scène pour devenir possiblement conscient (Hohmeyer, 2006). Au regard des traitements et des informations qu'elle fournit à la scène, l'inconscience joue un rôle important dans la construction du conscient. Par ailleurs, les modules inconscients vont réagir exclusivement à ce qui est conscient, excepté pour des processus automatisés, qui ne requièrent pas l'utilisation de la conscience pour s'exécuter. Il existe donc une forte intrication entre conscience et inconscience et c'est à travers ce jeu d'interdépendance que Baars imagine le fonctionnement cérébral humain.

En conclusion, plusieurs architectures simulant certains phénomènes de la conscience ont été présentées. L'agent conscient (CTS) que nous proposons se base essentiellement sur la théorie de Baars et a été construit à partir de IDA. Dans le prochain chapitre, je décris en

détails CTS en m'appuyant largement sur la théorie de Baars. Cette description me permettra de mettre en avant les forces et faiblesses de CTS.

CHAPITRE IV : L'AGENT CTS

Comme décrit précédemment, l'agent IDA présente des avantages certains. Il s'appuie sur un modèle rigoureux et tire partie des nombreux phénomènes de la conscience. L'architecture d'IDA a été conçue de sorte qu'elle soit polyvalente et s'utilise dans divers autres champs d'application. Moyennant quelques modifications, elle peut s'utiliser pour le développement d'un agent tuteur intelligent.

Un effort a été entrepris dans ce sens par les membres du laboratoire GDAC. C'est ainsi que Patrick Homeyer⁵, sous la direction du professeur Roger N'kambou s'est investi pendant près de deux ans à la réalisation informatique de CTS. Cette réalisation a ensuite été utilisée dans la thèse de Daniel Dubois (Dubois, 2007). Le mémoire de maîtrise de Patrick (Hohmeyer, 2006) décrit en détail l'ensemble de l'architecture qu'il a implémentée et présente les avantages de CTS par rapport à d'autres systèmes phares tels que BlackBoard, BDI, INTERRAP et ACT-R.

Comment fonctionne CTS? Quelles sont ses forces et limites? Tel est l'objectif de ce chapitre. Les limites, dont je fais part, font référence à celles retrouvées dans le système quand je l'ai repris au début de mon séjour au GDAC. Celles-ci vont servir de point de départ pour tout le reste du mémoire.

4.1 Architecture du CTS

L'agent CTS ressemble beaucoup à l'agent IDA⁶. Il en partage globalement l'architecture, les mêmes modules et le même fonctionnement (voir Figure 4.1 : Architecture de CTS). CTS a

⁵ Étudiant de maîtrise en informatique à l'UQAM (2004-2006).

⁶ Il y a eu accord entre le laboratoire Memphis et le laboratoire GDAC pour le partage du code de IDA.

L'agent CTS peut être partagé conceptuellement en trois parties distinctes : la Perception, le Théâtre (d'après le concept de Baars) et l'exécution d'actions. La Perception permet au système de traduire en un formalisme spécifique ce qui se passe dans son environnement. Le Théâtre gère l'information provenant des différents modules composant l'architecture. Finalement, l'exécution d'actions se fait dans une structure complexe qu'est le Réseau des Actes (RA) (inspiré de Maes, 1989, 1991). Cette dernière octroie au système la capacité de décider des actions à prendre. Ces trois parties communiquent entre elles, s'échangent de l'information et collaborent afin de mener l'ensemble du système vers un but commun. L'ensemble de ces trois parties interagissent de la manière suivante (voir Figure 4.2 : Vue conceptuelle générale de CTS) :

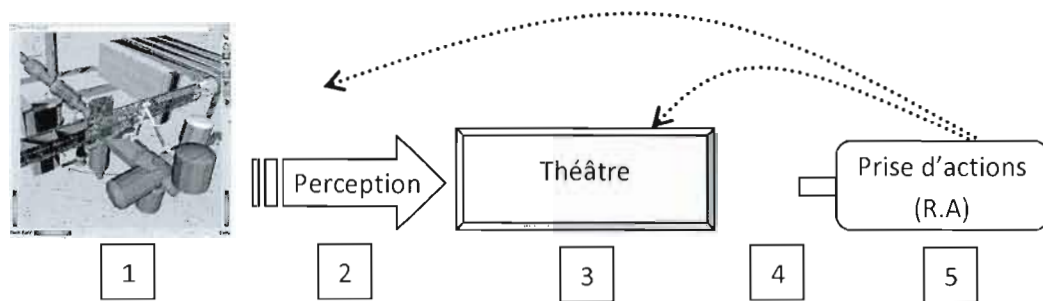


Figure 4.2 : Vue conceptuelle générale de CTS

Les informations externes qui parviennent à CTS sont issues du RomanTutor (étape 1). Ce dernier simule le panneau de commande virtuel du Bras canadien CANADARM2 et consigne toutes les actions de l'utilisateur relatives à la manipulation du Bras, de l'utilisation des caméras et de la détection de proximité avec l'un des modules de la SSI.

La Perception (étape 2) est responsable de la conversion de tout ce qui se passe dans le simulateur en un formalisme compréhensible par l'agent. Lors de la manipulation du RomanTutor, un message est transmis à CTS. La Perception analyse le message syntaxiquement et le traduit en *microprocessus d'information*. Ces derniers sont des entités élémentaires de stockage de l'information.

Ces microprocessus d'information issus de la Perception se retrouvent dans le Théâtre (étape 3) qui est responsable des traiter les informations. Il lui est possible de supprimer, créer, enrichir, modifier, prioriser, connecter et sélectionner l'information. Le Théâtre est composé de sous-modules : Mémoire de travail (scène selon Baars), module Conscience d'accès (*les acteurs sous le projecteur de lumière*), Modèle de l'apprenant (*les spectateurs*), Module du domaine (*les spectateurs*), microprocessus de raisonnement (*les spectateurs* et certains *opérateurs contextuels*) et finalement deux mémoires (*les spectateurs*)⁷. Tous ces éléments communiquent entre eux à intervalle régulier suivant un cycle bien précis (voir Figure 4.3 : **Organisation des sous modules composant le théâtre.**). Je reviendrai plus en détail sur la nature de ce cycle dans la section 4.3, mais en attendant il est important de garder à l'esprit que le Théâtre est composé de sous modules qui agissent de manière ordonnée.

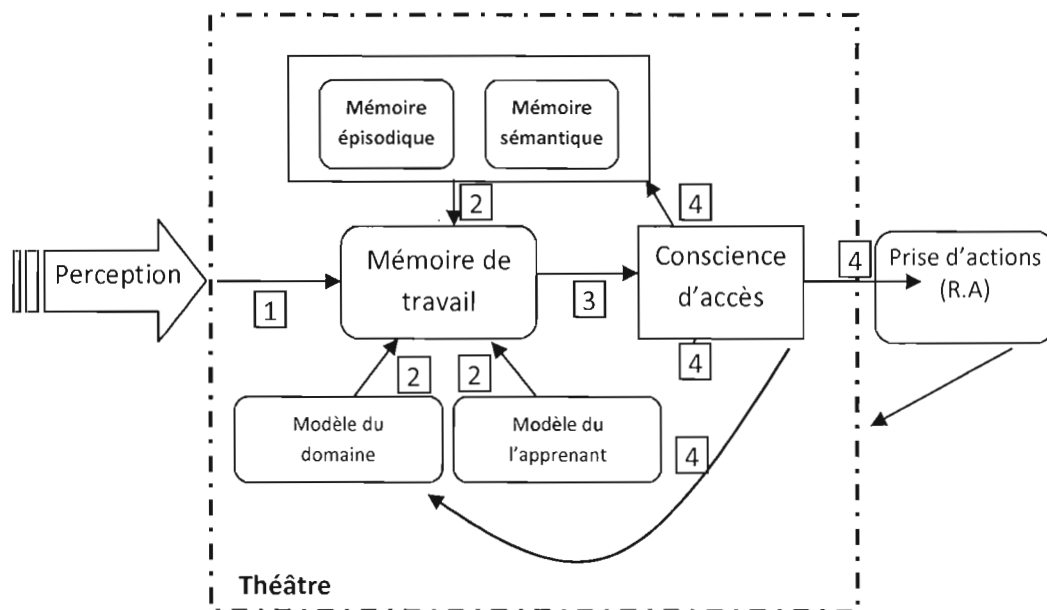


Figure 4.3 : Organisation des sous modules composant le théâtre.

⁷ Il existe en dehors du théâtre le RA. Il contient les buts et désirs qui représentent dans la théorie de Baars le directeur de scène.

Une partie de l'information traitée est reconnue par le RA qui a pour rôle de prendre la majorité des actions. C'est une structure exécutive évoluée qui renferme les actions didactiques/pédagogiques possibles que peut réaliser l'agent tuteur. Ces actions sont soit externes et donc visibles par l'apprenant, soit internes au système et donc invisibles. Les actions internes s'exécutent dans le cas où le RA nécessite des informations supplémentaires pour réaliser une action élaborée.

4.2 Modules, traitement et communication

Les modules qui composent l'architecture sont nombreux et interagissent de manière cyclique suivant un ordre précis. Tous les traitements se font à l'aide de **microprocessus** spécialisés. Seul le Module du domaine n'obéit pas à cette logique parce qu'il utilise le système de représentation développé et fourni par Fournier-Viger (2006).

Le module de la Perception réalise son traitement à l'aide des **microprocessus de Perception**. Ce module reçoit de RomanTutor des messages textes déterminés, spécifiés par une grammaire formelle. Un analyseur syntaxique transforme le message en arbre syntaxique inspecté par les microprocessus de Perception (Hohmeyer, 2006). Chaque microprocessus s'intéresse à une information particulière et décide de l'importance de celle-ci en fonction de trois attributs : la gravité, le changement et l'évolution⁸. Une fois la valeur de ces paramètres déterminés, le microprocessus de Perception crée un microprocessus d'information et lui donne un attribut, une valeur et une activation comprise entre [0,1] traduisant l'importance de l'information. Le microprocessus d'information ainsi créé est généralement lié à d'autres microprocessus. Le module de Perception est celui qui dicte s'il y a connexion avec d'autres microprocessus d'information. Par la suite, il est déposé dans la Mémoire de travail pour compétitionner et devenir

⁸ Par évolution, je fais référence aux situations qui se transforment positivement ou négativement dans le temps. Pour plus d'informations je vous invite à lire le mémoire de (Hohmeyer, 2006).

possiblement conscient. Voici un exemple explicite qui reproduit ce processus de traduction :

Tableau 4.1 : Traduction de l'information

Événement	Message issu du simulateur	Microprocessus d'information crée
L'apprenant manipule le Bras qui s'approche dangereusement d'un module de la SSI.	"this caused the danger of a collision between robot:WE and element S1P1TrussRight01 (distance : 6.95931)"	<pre> graph TD Collision["Collision : possible. (0.85)"] --> Distance["Distance : 6.95931 (0.4)"] Collision --> StationElement["StationElement : S1P1Truss (0.3)"] Collision --> Component["composant: WE (0.4)"] Component --> Position["Position : 0.0.0.0 (0.1)"] Component --> Canadarm["Canadarm : Canadarm (0.2)"] </pre>

La Mémoire de travail est un bassin qui collecte toutes les informations issues de la Perception, du Module du domaine, du Modèle de l'apprenant, des mémoires et du RA. Exception faite de la perception, tous les microprocessus d'informations présents dans la mémoire de travail sont dus à une réaction des autres modules à ce qui a été publié par la conscience lors des cycles précédents.

Des informations peuvent aussi provenir des **microprocessus de raisonnement**. Périodiquement, tous les microprocessus d'information présents dans la Mémoire de travail perdent progressivement leur activation. Cette implémentation s'appuie sur le fait que plus une information est ancienne, plus elle est quelconque en accord avec la théorie de Jackson (1987). De ce fait, tous les microprocessus d'information qui se retrouvent dans la Mémoire de travail y restent jusqu'à ce qu'ils soient publiés ou qu'ils perdent progressivement leur activation pour finalement disparaître. Pour qu'une information soit publiée, il faut au préalable qu'elle fasse partie d'une **coalition** gagnante.

Une coalition est un groupe de microprocesseurs d'information liés entre eux par des liens orientés. Le microprocesseur d'attention qu'est le CoalitionManager est celui qui crée ces coalitions. La figure suivante expose les coalitions possibles qui peuvent se former dans le cas d'une information qui décrit une collision :

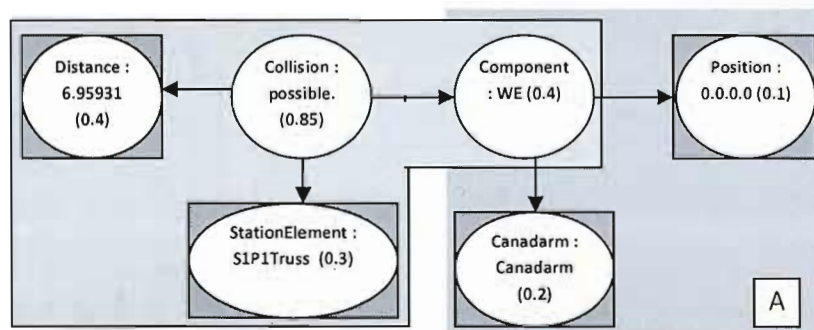


Figure 4.4 : Type de coalition possible : les 6 rectangles représentent les coalitions possibles

Le CoalitionManager sélectionne les microprocesseurs d'information un à un et crée une coalition en parcourant les enfants immédiats du microprocesseur central présents dans la mémoire de travail (exemple : nœud Position est le fils du nœud Component). Une fois les coalitions formées, vient ensuite le SpotLightController qui a pour rôle de calculer la valeur de la coalition nouvellement formée. Le calcul se fait en effectuant la somme de la valeur du nœud central avec la moyenne de celle de ses fils. Pour la coalition A, la valeur est de : $((0.1+0.2)/2)+0.4 = 0.55$. Ce genre de calcul est intentionnel et permet de :

1. Favoriser la publication de coalitions composées de plusieurs microprocesseurs d'information.
2. Faire en sorte que l'activation du microprocesseur formant le centre de la coalition soit plus importante que l'ensemble des autres microprocesseurs.

La coalition dont l'activation est la plus forte gagne la compétition. Elle est publiée à l'aide du BroadcastManager qui se charge de l'extraire de la Mémoire de travail et de la rendre consciente, c'est-à-dire connue par l'ensemble des modules qui composent le Théâtre et le RA.

Le RA est l'entité responsable de la sélection et du déclenchement d'une action donnée. Développé dans les travaux de Pattie Maes (Maes, 1989), le RA se compose d'Actes liés formant un réseau d'actions. Afin de hiérarchiser le Réseau des Actes, Negatu et Franklin (1999) l'ont enrichie avec des « Drives » (que je traduis en *Désirs*) qui forment les buts ultimes de l'agent et les « Goals » qui sont des buts intermédiaires (voir Figure 4.5 : Une séquence rudimentaire du Réseau des Actes.).

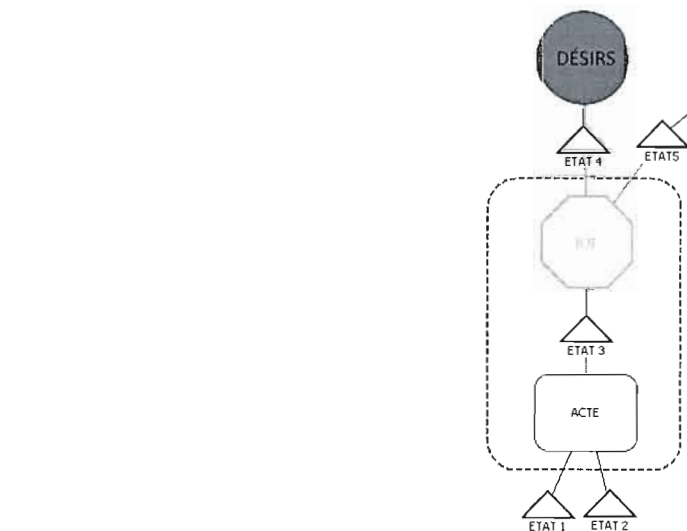


Figure 4.5 : Une séquence rudimentaire du Réseau des Actes.

Le lien entre les Actes du réseau se fait par l'entremise d'États. Ces derniers sont connectés aux Actes à l'aide de trois types de connexions différentes : inhibition, précédence et antécédence. Ces connexions permettent de structurer le réseau et aussi de faire circuler

l'énergie en respectant les règles imposées par cette structure. L'énergie d'activation qui circule dans le réseau prend son origine des *Désirs* ou des États vérifiés (qui sont à vrai). L'énergie circule d'Acte en Acte en suivant les trois types de liens qui au final inhibent ou augmentent l'activation. Quand un Acte a tous ses États à vrai et dispose du plus haut niveau d'activation, par rapport à tous les autres Actes du réseau, alors il **s'exécute**. Une fois déclenché, tous les **microprocessus d'actions** qu'il contient s'exécutent à leur tour (voir Figure 4.6 : Exemple générique d'un Acte.). Ces microprocessus d'action peuvent être de deux types : moteur (action externe; exemple = rejouer une séquence de manipulation du CANADARM II) ou raisonnement (action interne; exemple = déterminer la cause de l'inactivité de l'apprenant). Ces microprocessus d'action n'existaient pas dans le réseau originel de Maes, ils ont été ajoutés par Franklin (1998) afin de connecter cette structure décisionnelle au reste de l'architecture, qui se veut soutenue essentiellement par des microprocessus.

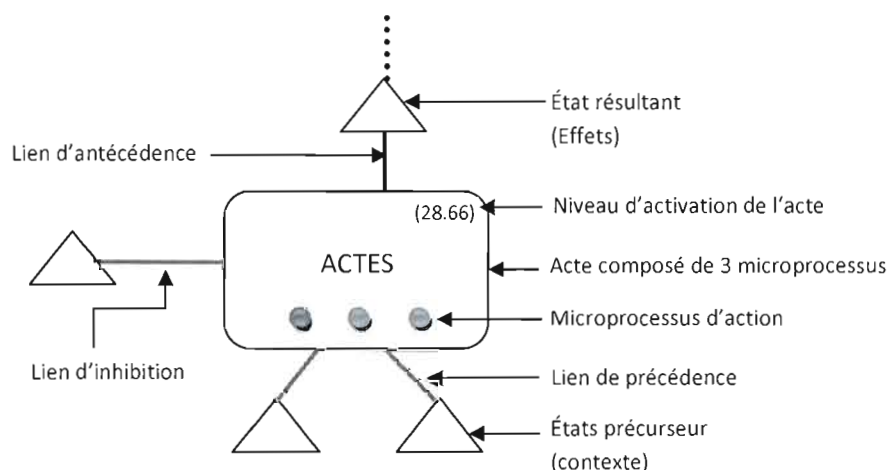


Figure 4.6 : Exemple générique d'un Acte.

Il est possible de voir le RA comme une base de règles disposant d'un paramètre discriminant qu'est l'activation de l'Acte. Cette particularité permet au RA d'être plus intéressant qu'une simple base de règle et lui confère la capacité de planifier à l'avance ses

actions (Decugis et Ferber, 1998). En fait, dans tout le réseau, certains Actes, en fonction de leur connexion aux autres Actes, des États qui les composent et des Désirs qui lui sont connectés, vont accumuler beaucoup plus d'activation que d'autres. Au final, l'Acte peut accumuler beaucoup d'énergie, ce qui favorise son exécution par rapport à d'autres. Grâce à l'accumulation progressive d'énergie dans les Actes, le RA effectue une analyse plus globale, multicritères et temporelle, qui ne suit pas simplement l'état des faits actuels, mais aussi se montre sensible aux événements antérieurs (Gaha, Dubois et Nkambou, 2008).

4.3 Cycle cognitif

Tous les éléments constituant l'architecture de CTS obéissent à un fonctionnement cyclique et répétitif. Ils interagissent de manière ordonnée et ponctuelle en suivant un plan d'action prédéterminé qu'est le *cycle cognitif*. Ce dernier vient englober tout ce qui a été dit précédemment et sous-tend la logique de fonctionnement de toute l'architecture. Le cycle de CTS se compose de huit étapes qui sont constamment répétées.

1. **Perception** : les messages provenant du micro monde (dans notre cas RomanTutor) sont traduits et transformés en microprocessus d'information.
2. **Transfert en Mémoire de travail** : le stimulus, tel qu'interprété par la Perception, se voit déposé dans la Mémoire de travail. Toute autre source d'information interne (Modèle de l'apprenant, Module du domaine, etc.) dépose aussi ses apports spontanés en MT à cette étape.
3. **Rappel des informations associées** : les Mémoires à moyen et long-terme reçoivent une copie des informations entrées en Mémoire de travail. Elles retournent des

informations apparues dans le passé, complétant ainsi une information parfois partielle⁹.

4. Sélection des coalitions

4.1. **Raisonnement par les microprocessus de raisonnement** : tous les microprocessus de raisonnement s'exécutent. Ils sont capables de prendre des actions, de rechercher une information particulière dans la Mémoire de travail, de créer, s'opposer à, ou supprimer des microprocessus d'information qui proviennent d'autres modules, *etc.*

4.2. **Compétition pour la conscience** : lors de cette étape, il y a formation des coalitions à partir des microprocessus d'information, et calcul des valeurs de chacune des coalitions.

5. **"Publication" de la coalition gagnante** : la coalition gagnante quitte la Scène pour se retrouver dans le module de Conscience d'accès. Tous les modules et microprocessus de raisonnement reçoivent la publication de la coalition gagnante. L'information est dupliquée et devient connue par tous. Une fois publiée, celle-ci quitte à son tour la Conscience d'accès.

6. **Recrutement de ressources** : tous les modules et les microprocessus qui reconnaissent l'information de la coalition publiée vont réagir. Contrairement à l'étape 2 ces ressources réagissent uniquement à la publication. Rappelons que ces derniers peuvent réagir sur l'absence d'une information particulière. En ce qui concerne le RA, certaines des informations publiées mettent à Vrai ses États. Cette étape amorce l'exécution d'actions et met fin à la phase de raisonnement et de traitement de l'information.

⁹ Cette étape du cycle ne sera pas abordée dans mon mémoire. Pour plus de détails sur son mode de fonctionnement veuillez vous référer aux travaux de Faghihi Usef (2007).

7. **Choix de l'Acte à prendre** : les États mis à Vrai par l'étape 6 ainsi que le niveau d'activation de l'Acte font en sorte qu'il s'exécute parmi plusieurs autres. Il arrive qu'aucun Acte ne puisse être exécuté, ce qui signifie que le système ne dispose pas des préconditions nécessaires ou que l'Acte n'a pas l'énergie suffisante pour s'exécuter.
8. **Exécution de l'Acte choisi** : si un Acte est choisi, il est déclenché. Cela aboutit à l'exécution des microprocessus d'action que l'Acte contient.

4.4 Forces de CTS

L'architecture CTS présente des avantages dont j'énumère ici les plus importants :

- **Traitement des informations** : le système réalise à l'aide de la Mémoire de travail et de la Conscience d'accès un filtre sur l'information à traiter. À l'aide de la valeur d'activation de l'information, CTS parvient à prioriser l'information la plus pertinente pour être soumise et éventuellement traitée par toutes les ressources composant l'architecture. Le système évite ainsi une explosion du nombre de propositions possibles et, par conséquent, la saturation du système. Ce travail de collaboration entre la Mémoire de travail et la Conscience d'accès permet de traiter une information à la fois par plusieurs ressources.
- **Théorie de Baars et le projet IDA** : l'agent CTS tire sa force de deux éléments complémentaires. En premier lieu, CTS utilise les préceptes de la théorie de Baars pour modéliser le fonctionnement du système. Cette affiliation octroie à l'agent une assise scientifique en ce qui concerne le fonctionnement cognitif. D'autre part, l'équipe de Franklin a collaboré avec l'équipe du GDAC et a fourni une partie du code d'IDA. Une grande part du code de CTS s'inspire de celui d'IDA quoique l'intégralité de l'implémentation de CTS reste un travail propre à l'équipe du GDAC.
- **Reproduction de certains phénomènes conscients** : l'architecture de CTS s'appuie sur les préceptes de Baars, ce qui lui confère la capacité de simuler certains des

phénomènes liés à la conscience. En premier lieu, l'agent déploie un premier niveau de conscience dans la traduction de son environnement. Deuxièmement, CTS reproduit le phénomène de la *conscience d'accès*, qui permet d'interconnecter des ressources inconscientes de l'architecture, autonomes les unes des autres. Troisièmement, à l'aide de la Mémoire de Travail, l'agent agglomère et priorise les informations pour les traiter selon un ordre d'importance. Par exemple, une information qui exprime le danger d'une proximité du Bras CANADARM2 est traitée avant toutes les autres. Finalement, l'agent est en mesure de résoudre des problèmes à l'aide de la délibération ainsi qu'éditer des plans. Ces deux dernières fonctionnalités seront abordées dans le Chapitre Six « Émotion lors du tutorat ».

- **Indépendance entre les modules** : l'implémentation de CTS est réalisée de sorte à permettre une indépendance des différents modules composant l'ensemble de l'architecture. En effet, chaque élément est développé, dans un package Java indépendant (voir Figure 4.7 : Organisation des packages de CTS). Cette approche octroie au système une grande modularité et permet par de remplacer aisément le module de la Perception et d'appliquer le système à d'autres domaines.

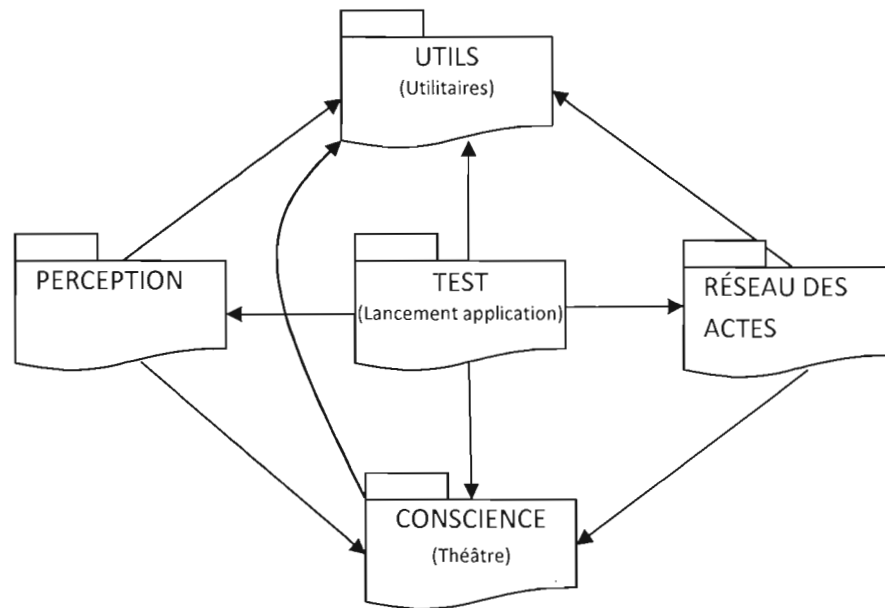


Figure 4.7 : Organisation des packages de CTS

- Polyvalence de l'architecture :** l'architecture de CTS vise à reproduire les mécanismes cérébraux humains. Cette tentative de reproduction octroie au système une grande polyvalence d'action. Même si CTS s'est voulu spécialisé dans l'enseignement, il aurait pu s'utiliser dans d'autres tâches telle l'affectation des postes pour les marins (Franklin 1998) ou a un système de triage des patients d'un hôpital (Franklin et Jones, 2004). Tout comme CLARION ou GLAIR, l'agent CTS est théoriquement capable de reproduire une grande majorité des tâches que l'Homme est capable de faire, car il cherche à reproduit au mieux le fonctionnement cérébral humain.
- Exécution d'actions ingénieuse :** le fait d'utiliser le RA pour modéliser les actions du tuteur offre une grande souplesse au système. En effet, ce réseau est capable de planifier certaines séquences d'Actes à l'avance. L'action exécutée par le réseau est celle qui a toutes ses préconditions satisfaites et dont le niveau d'énergie est le plus

haut. Ainsi, le RA va favoriser l'exécution de l'action la plus opportune, celle qui paraît la plus pertinente ou qui se déploie en un minimum d'étapes. Le comportement du RA se veut émergent et faiblement déterministe (Maes, 1989). Cette caractéristique permet au tuteur de réagir de manière changeante grâce à un même RA.

- **Édition du Réseau des Actes :** le RA est un élément important dans l'agent CTS, car toutes les actions pédagogiques et didactiques prises passent par cet élément. Il devient essentiel de pouvoir développer rapidement et facilement pareil réseau, parce que celui-ci est appelé à être constamment enrichi tout au long de la construction de CTS. C'est pour cette raison qu'un éditeur de RA a été développé en tant que *plugin* sur la plateforme Eclipse 3.1. Celui-ci peut être utilisé par des non-informaticiens car il permet de construire graphiquement et rapidement un réseau complexe. L'éditeur génère par la suite un fichier en XML qui est par la suite chargé dans le moteur de CTS pour être lu et converti en objet java. Ces derniers vont servir tout au long du cycle cognitif. Voici à quoi ressemble l'éditeur en question.

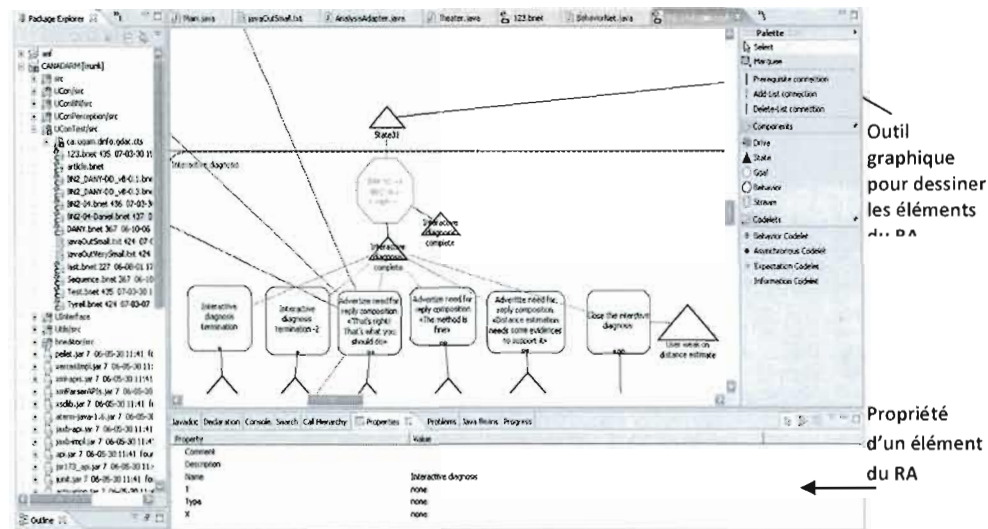


Figure 4.8 : Éditeur du RA

4.5 Faiblesses de CTS

Tous les systèmes informatiques ont des forces et des faiblesses. Ce constat est d'autant plus vrai pour CTS qui reste une architecture jeune. Celle-ci a encore besoin de mûrir et sa mise en œuvre requiert une poursuite intense de l'effort pour atteindre un degré de fiabilité acceptable. Voici les limites décelées au moment où j'ai repris la mise en œuvre de CTS:

- **Absence d'interaction :** l'agent CTS est dans l'incapacité de communiquer avec l'apprenant ou le RomanTutor et aucune interface ou fenêtre de communication n'était implémentée. CTS se limite à effectuer des traitements en interne qui, au final, ne sont pas visibles par l'apprenant. D'autre part, CTS n'est pas en mesure de profiter des fonctionnalités offertes par RomanTutor qui joue le rôle d'Expert du Domaine. En fait, il est impossible au tuteur de faire rejouer des séquences vidéo, de faire exposer des exercices à réaliser par l'apprenant ou de faire planifier des trajectoires, etc. La communication entre RomanTutor et CTS est unidirectionnelle et seuls les messages issus de la manipulation du Bras CANADARM2 sont perçus et traités par CTS. Il y a lieu de créer un pont de communication entre CTS (développé en Java) et RomanTutor

(développé en C++) pour pouvoir commander l'environnement virtuel. De plus, il faut permettre au système de communiquer avec l'apprenant à l'aide d'interfaces et de fenêtres de dialogue.

- **Modèle de l'apprenant et Module du domaine inexistant** : l'une des plus importantes limites du système réside dans l'inexistence des Module du domaine et de l'apprenant. Aucun élément dans l'architecture telle que fournie par Hohmeyer n'implémente ou ne simule ces deux modules, qui pourtant sont indispensables pour l'agent tuteur. Ces derniers sont nécessaires pour dispenser du contenu ainsi que de fournir un cours personnalisé. Il y a lieu d'intégrer ces deux éléments et de les faire réagir en synchronisation avec le cycle cognitif de l'agent.
- **Réseau des Actes pauvres** : ce dernier est très peu développé, il n'offre qu'un seul scénario fonctionnel. Ce réseau permet seulement de **simuler** une délibération qui diagnostique la cause d'une proximité (voir Figure 4.9 : Scénarios de délibération). La finalité du scénario est beaucoup plus de démontrer que le RA réagit et fonctionne bien. Ce scénario est loin d'être suffisant car il retourne toujours le même résultat et l'agent CTS doit obligatoirement disposer d'une plus grande richesse d'action. Le RA doit être enrichi afin de rendre le tuteur réactif, capable de prendre des actions pédagogiques pertinentes, variées et réalistes.

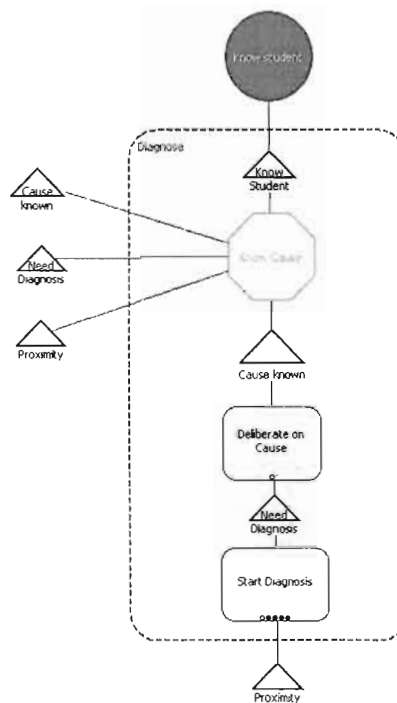


Figure 4.9 : Scénarios de délibération

- Fonctionnement du Réseau des Actes :** c'est est une structure largement utilisée dans différentes applications de pointe telles que la planification et la robotique (Matsuura et Wada, 2000; Luo, Jiang, et Zhu, 2004; Maes, 1991). Il n'en demeure pas moins que plus le réseau est étendu et riche, plus il devient difficile à « équilibrer » en énergie. Il commence à réagir de manière chaotique et inattendue. En effet, les équations qui gouvernent la diffusion de l'énergie du RA deviennent rapidement incomplètes quand il s'agit d'un réseau composé de nombreux Actes, liens et États. Ces problèmes ont été soulevés dans l'excellent travail de Tyrell (1994) qui met en évidences les incomplétudes et les faiblesses du réseau de Maes. En effet, trouver le moyen d'enrichir les équations qui gouvernent la diffusion de l'énergie dans le réseau pour produire un réseau étendu et équilibré est primordial.

- **Équilibrage de toute l'énergie :** une autre difficulté inhérente à l'architecture particulière de CTS réside dans l'interaction entre le Théâtre et le RA. D'un côté, dans le Théâtre, les microprocessus d'information ont des valeurs d'activations changeantes en fonction du temps passé dans la Mémoire de travail et de la valeur intrinsèque de l'information ; les valeurs sont comprises entre $[0,1]$. De l'autre, le RA contient lui aussi de l'énergie qui circule dans les États en fonction d'équations mathématiques différentes de celles du Théâtre; les valeurs y fluctuent dans l'intervalle $[0,80]$. Or, ces deux modules sont appelés à partager des microprocessus d'informations et de l'énergie tout au long du cycle cognitif. Il y a lieu de trouver le moyen de faire interagir ces deux mondes de sorte qu'au final il s'y installe un équilibre. Dans CTS, ce phénomène a été trop peu analysé et une étude plus poussée serait nécessaire.
- **Agrégation limitée :** l'agrégation permet de créer les coalitions de microprocessus d'information et au bout du compte de spécifier quelles sont les informations à être publiées en priorité par la Conscience d'accès. Or dans le système actuel, seules des agrégations d'un seul niveau sont possibles. C'est-à-dire qu'une coalition ne peut pas être composée de plus d'un niveau d'information (voir Figure 4.10 : **coalitions limitatives**). Cette manière de faire a été initialement implémentée afin de faciliter la création des coalitions et leur manipulation. Toutefois, cette approche limite considérablement l'expressivité de la coalition et il arrive que la Conscience traite une information biaisée et incomplète. Pour pallier cette limite, il faut trouver le moyen de créer des coalitions plus expressives, composées de plusieurs niveaux d'informations.

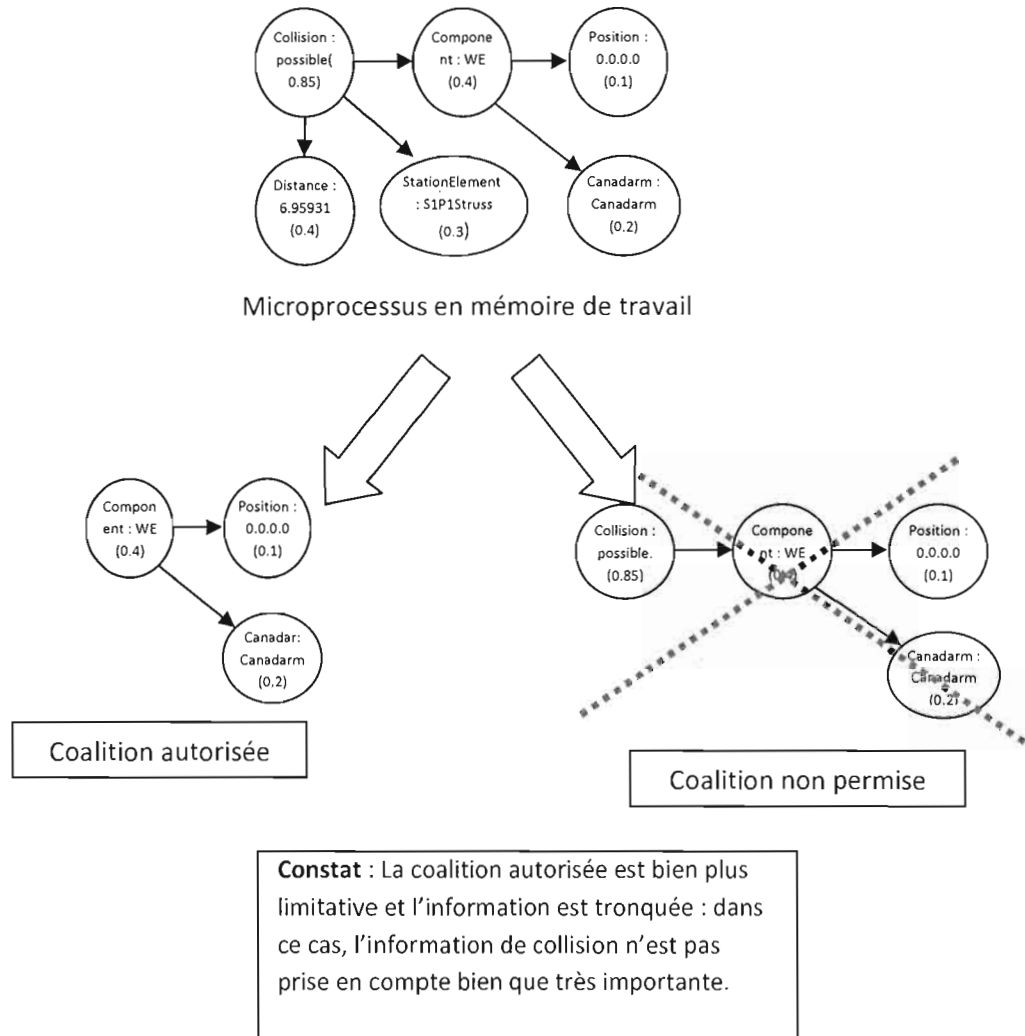


Figure 4.10 : coalitions limitatives

- Les émotions et le tutorat :** le rôle des émotions lors du processus d'apprentissage est de plus en plus évident. De nombreuses recherches font ressortir leurs relation (Châffar et Frasson, 2004 ; O'Regan, 2003 ; Hara et Klan, 2000). En effet, les émotions, qu'elles soient positives ou négatives, ont un impact sur les processus cognitifs de l'apprenant, et ce, à différents niveaux (Lisetti et Nasoz, 2004). En tenir compte s'avère important

pour garantir un enseignement qui s'ajuste au mieux aux besoins de l'apprenant. Permettre au tuteur de tenir compte des émotions afin de réagir plus finement face à l'apprenant donne au système un comportement proche d'un véritable enseignant humain. Or, CTS ne tient nullement compte de l'aspect émotions et ne dispose pas des fonctionnalités informatiques pour les capter et les traiter. Étendre CTS de sorte qu'il soit en mesure de réagir aux émotions de l'apprenant constitue une avancée importante pour ce tuteur.

- **Les mémoires :** l'agent LIDA (Learning IDA) (Ramamurthy *et al.*, 2006) est une extension d'IDA qui introduit principalement le concept d'apprentissage. Des mémoires spécialisées ont été introduites dans LIDA, afin de permettre à l'agent d'apprendre. Ces mémoires existent conceptuellement dans l'architecture de CTS, mais du point de vue implémentation, il reste beaucoup de travail à faire. Pour le moment, l'agent n'est pas capable d'apprendre de ses Actes, et à chaque fois que le système s'exécute, il reproduit les mêmes actions comme s'il s'agissait de la première fois. Introduire des mémoires qui enregistrent certains résultats ou conclusions réalisés à l'aide de l'apprentissage machine permettrait d'économiser un précieux temps de calcul.
- **Manque de clarté du fonctionnement :** le théâtre de l'agent CTS se compose de plusieurs modules qui créent, associent, réagissent et suppriment de l'information. Parallèlement, le RA crée à son tour de l'information qui se répercute sur tous les modules. Ces deux éléments interagissent en continu via le cycle cognitif. C'est pourquoi il devient vite difficile de savoir au bout de quelques itérations quelles sont les actions de chacun des modules. Or, il n'existe qu'une fenêtre simpliste qui fait état des activités de l'agent CTS (voir Figure 4.11 : **Fenêtre du théâtre**). Ainsi, il devient très difficile de savoir comment évolue le système. Pourtant, comprendre le fonctionnement du système et déterminer les actions de chacun des éléments de l'architecture sont importants pour toutes actions de débogage futures. Un tableau de bord qui affiche l'évolution du système, la manière dont transite l'information ainsi que les transformations du RA devient plus qu'indispensable.

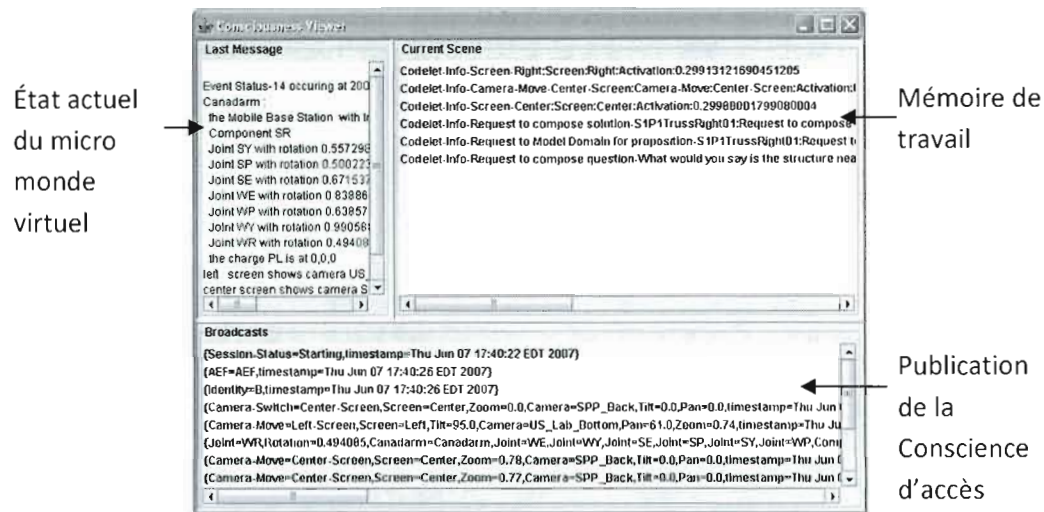


Figure 4.11 : Fenêtre du théâtre

En conclusion, CTS reste un projet encore très jeune et plusieurs modules et fonctionnalités restent à développer. Il y a lieu de fournir un effort important afin de rendre l'agent fonctionnel et capable de réaliser des actions pédagogiques. De plus, tel qu'exposé dans la figure ci-après, seul un petit nombre de modules ont été intégrés jusqu'à présent.

Comme CTS est une architecture au comportement émergent, le fait de rajouter des modules augmente la complexité de l'architecture. En effet, un petit changement opéré sur l'architecture modifie considérablement le fonctionnement de tout le système. Des

discussions faites dans ce sens avec Franklin¹⁰ confirment cette appréhension. Il y a lieu de développer CTS de manière incrémentale.

¹⁰ Au cours d'une conversation avec le professeur Stan Franklin lors de sa visite à l'UQAM en Mai 2007

CHAPITRE V : APPORTS ET MODIFICATIONS DE CTS

Afin de rendre l'agent CTS fonctionnel et le doter de réactions pédagogiques efficaces, des modifications à différents niveaux de l'architecture sont à réaliser. Dans ce chapitre, je présente les différents ajouts et modifications que j'ai réalisés sur l'agent CTS pour aboutir à un agent fonctionnel doté de réactions pédagogiques et didactiques adéquates.

5.1 Communication CTS/Apprenant

Lorsque j'ai repris l'architecture de l'agent CTS, celle-ci disposait d'une réactivité très limitée et aucune interaction avec l'utilisateur n'était possible. Je me suis mis à réaliser un module supplémentaire capable de communiquer avec l'utilisateur de manière bidirectionnelle et centralisée dans un tampon de sortie. Pour ce faire, j'ai implémenté un module additionnel de communication qui réalise deux types d'interactions, à savoir afficher (1) un message informatif ou (2) une interrogation qui nécessite une réponse de la part de l'utilisateur (voir Figure 5.1 : Exemple de fenêtres informative et interrogative).

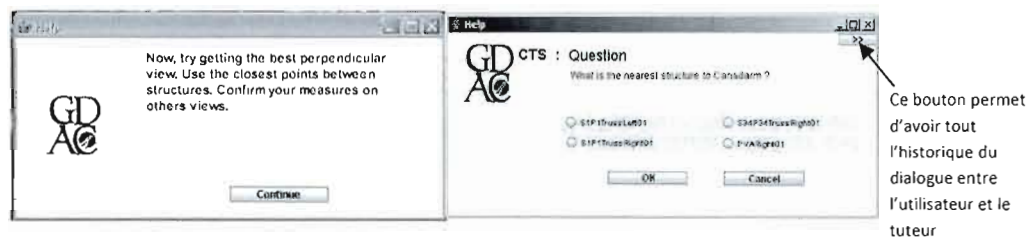


Figure 5.1 : Exemple de fenêtres informative et interrogative

La mise en œuvre de la décision d'afficher ces fenêtres se fait quand un Acte d'affichage s'exécute et lance ses microprocessus d'action. Ces derniers créent des microprocessus d'information qui spécifient toutes les propriétés du type d'intervention à réaliser (nature de la fenêtre, message à afficher, propositions ou choix possibles, boutons...).

Dans certains cas, il arrive que les informations à afficher à l'utilisateur dépendent du contexte de manipulation du Bras robotisé. Pour résoudre ce problème, j'ai adopté l'approche de Rhodes (1995) qui insère des variables additionnelles dans les Actes pour rendre spécifiques des actions génériques. Rhodes prend l'exemple d'un Acte « *prendre outil* » et associe à l'attribut « *outils* » une variable (exemple = marteau), ce qui transforme l'Acte en « prendre marteau ».

Dans CTS, j'ai fait en sorte que les États jouent le rôle d'attributs en gardant en mémoire la coalition qui les a mises à vrai. En permettant aux microprocessus d'action contenus dans les Actes d'accéder à ces coalitions, ils deviennent capables d'affecter des valeurs aux attributs. Afin de créer des microprocessus d'information au contenu variable, j'ai doté les microprocessus d'action d'un petit analyseur lexical capable d'extraire l'information à partir des coalitions. Le tableau suivant expose le déroulement de ces actions :

Tableau 5.1 : Exécution d'un Acte d'affichage

Étape	Opération réalisée	Lors de l'exécution
1	<p>Une coalition d'information (A-B-C) vient mettre à Vrai l'État (1) d'un Acte d'affichage. La coalition contient des microprocessus informant du contexte dans lequel s'est produite la collision.</p> <p>(Lieu : Réseau des Actes)</p>	
2	<p>Exécution du microprocessus AC de l'Acte d'affichage. Ce microprocessus extrait la variable de l'attribut « component » pour créer un microprocessus d'information.</p> <p>(Lieu : Réseau des Actes)</p>	
3	<p>Le microprocessus d'action crée une information et la dépose dans la Mémoire de travail avec l'information correspondante.</p> <p>(Lieu : Mémoire de travail dans le Théâtre)</p>	

Par la suite, ces microprocessus d'information sont déposés dans la Mémoire de travail et une fois entrés dans la « conscience », ils sont collectés par un microprocessus nommé *MessageComposition.java* qui joue le rôle de tampon de sortie. Il assemble les données d'affichage et décide quel type de fenêtre à afficher, comment remplir les différents champs de la fenêtre, les boutons à insérer, et la durée d'affichage de la fenêtre. Dans le cas où la fenêtre affichée à l'apprenant nécessite une interaction, la réponse de l'utilisateur est traduite sous forme de microprocessus d'information pour être déposée dans la Mémoire de travail. Voilà en somme comment s'opère la communication entre CTS et l'apprenant.

D'autre part, comme CTS se devait de communiquer avec RomanTutor afin d'en tirer profit (tracer des trajectoires, présenter des exercices, planifier des trajets, etc.), j'ai implémenté un pont de communication entre CTS et RomanTutor en utilisant des méthodes natives en C++. Il est dorénavant possible pour CTS de pouvoir exécuter **toutes** les fonctionnalités de RomanTutor.

5.2 Simulation de modèles

Pour que l'agent CTS dispense un cours, il doit pouvoir disposer d'un contenu tutoriel à présenter. En collaboration avec Philippe Fournier-Viger¹¹, j'ai établi un pont au Module du domaine qu'il a développé et adapté, dans le cadre de son projet de maîtrise, au domaine d'apprentissage des astronautes. Son modèle de représentation des connaissances est cognitif et s'inspire des travaux d'ACT-R pour représenter des connaissances spatiales (Fournier-Viger, Nkambou, et Mayers, 2007; Fournier-Viger *et al.*, 2006).

Par ailleurs, pour permettre au Module du domaine de s'intégrer à l'architecture de CTS tout en respectant les consignes de Baars, j'ai développé un microprocessus de raisonnement nommé *DomainModelListener* qui joue le rôle d'interface entre la Mémoire de travail et ledit modèle. Ainsi, à chaque cycle cognitif, le Module du domaine connaît tout ce qui est publié par la Conscience et peut créer des microprocessus d'information en Mémoire de travail. Le tableau 6.2 suivant expose la manière dont réagit le Module du domaine. En entrée, il reçoit les informations publiées par la Conscience. En sortie, il y a les microprocessus d'informations créés et déposés dans la Mémoire de travail. Les informations en entrée et en sortie sont plus complexes que ce qui est représenté dans le tableau, mais ces indications permettent d'avoir une bonne idée des interactions possibles.

¹¹ Étudiant au doctorat en Informatique cognitive à l'Université du Québec À Montréal.

Tableau 5.2 : Mode de communication du Module du domaine

Module du domaine		Exemple d'exécution du Module du domaine	
<i>Entrée</i>	<i>Sortie</i>	<i>Entrée</i> ¹²	<i>Sortie</i> ¹⁵
Demande de texte explicatif concernant un concept particulier (ex : définir « repère visuel »).	Texte explicatif concernant ce qu'est un repère visuel.	"Request to compose message on milestone Concept"	"Milestones are physical or virtual locations that are made part of..."
Demande pour fournir un conseil particulier dépendant d'un ordre de profondeur.	Conseil du niveau demandé visant à aider l'utilisateur lors de sa manipulation du Bras.	"Request to compose Hint level 2"	"Will you be able to evaluate distances properly from what you are seeing now?"
Détection d'une séquence de manipulation inappropriée par l'utilisateur	Information concernant l'étape omise.	[dépendant des actions de l'utilisateur]	"Have you not forgotten to change central view camera to evaluate the distance with more accuracy?"
Demande pour fournir des propositions pour une question à choix multiples.	Ensemble de propositions pour remplir les champs d'une question à choix multiples.	"Request to compose proposition on distance question for element S1P1TrussRight01"	"propositions = S1P1TrussRight01-- S1P1TrussLeft01-- S34P34TrussRight01--"

De la même manière, j'ai simulé un Modèle de l'apprenant élémentaire capable de reconnaître deux utilisateurs distincts (nommés A et B) dotés d'aptitudes et de défaillances différentes. Même s'il s'agit d'une simulation rudimentaire ad hoc, elle démontre néanmoins la faisabilité de l'approche. En effet, à l'image du Module du domaine, il est possible de développer un Modèle de l'apprenant séparément et, par la suite, l'intégrer

¹² Les entrées et les sorties dans le système sont uniquement des microprocessus d'informations.

dans CTS. Le tableau 6.3 expose les différentes manières dont opère le Modèle de l'apprenant.

Tableau 5.3 : Mode de communication du Modèle de l'apprenant

Modèle de l'apprenant		Exemple d'exécution du Modèle de l'apprenant	
<i>Entrée</i>	<i>Sortie</i>	<i>Entrée</i> ¹³	<i>Sortie</i> ¹⁷
Nom et mot de passe de l'utilisateur	Identification de l'utilisateur réussie ou échouée.	"Identify user name and password"	"Identification successful"
			"Identification failed"
Déterminer si l'utilisateur est bon en évaluation des distances ou pas.	Informe de la maîtrise de l'utilisateur en évaluation des distances.	"Is user weak on distance evaluation"	"User weak on distance evaluation"
			"User not weak on distance evaluation"
Déterminer pourquoi l'utilisateur a omis une étape lors de la manipulation du Bras.	Propose une ou plusieurs causes possibles à cette omission.	"Problematic situation, user missed a manipulation step"	"Cause is the poor mastery of manipulation with (60% confidence)"
Déterminer la préférence d'intervention pour l'apprenant	Informe que l'utilisateur préfère un média plus qu'un autre pour présenter un conseil.	"The cause is known and intervening is approved"	"Learner prefers hint"

5.3 Enrichissement des scénarios

Comme le RA ne disposait que d'un seul scénario, je devais y remédier en enrichissant l'éventail d'actions du Tuteur. La réalisation d'un RA plus élaboré s'est faite en plusieurs étapes à travers un cycle incrémental et itératif. En effet, avant d'arriver à la version

définitive du RA, j'ai dû passer par la création préalable d'un ensemble de sous réseaux¹⁴. Cette étape constitue une tâche difficile et délicate, car il est toujours laborieux dans ce genre d'implémentation de savoir quelles sont les bonnes suites d'actions à activer pour atteindre un but en particulier. De plus, il y a lieu de déterminer le lien entre les stimuli en entrée et les Actes à exécuter. Toutes ces difficultés se regroupent sous la bannière du *problème de sélection d'action* (Atkinson-Abutridy et Carrasco-Leon, 1999). Au final et en concertation avec certains membres du GDAC, j'ai élaboré cinq scénarios fonctionnels dont voici une description.

Premier et deuxième scénario : Ces deux premiers scénarios sont spécifiques à une situation particulière et exécutés uniquement dans le cas où il y a un risque de collisions avec l'une des structures de l'ISS. Ces deux scénarios se complètent, le premier détenant les préconditions nécessaires à l'exécution du second scénario.

L'Acte du premier scénario s'exécute dans le cas où il y a publication d'un microprocessus d'information faisant état d'une possibilité de collision. Cet Acte permet d'afficher une fenêtre interrogative et la réponse de l'apprenant sert de précondition à l'Acte suivant (voir Figure 5.2 : Portion du premier scénario du RA).

Ainsi, une série de questions est posée à l'apprenant et au fur et à mesure de ses réponses, certains États se mettent à vrai. Ces derniers vont servir de préconditions pour l'exécution du second scénario qui laisse place à des actions de type remédiation. Ce scénario a la délicate tâche d'expliquer à l'apprenant les causes de la mauvaise manipulation qu'il a effectuée.

¹³ Les entrées et les sorties dans le système sont uniquement des microprocessus d'informations.

¹⁴ Ces scénarios n'ont pu être possibles qu'après modification du mode de fonctionnement du R.A.

Quatrième scénario : Contrairement aux autres scénarios qui sont spécifiques, celui-ci est polyvalent et s'utilise dans différentes situations. Il fait intervenir tout le potentiel de l'architecture et permet au RA de jouer un rôle plus proche de l'idée originelle de Maes où l'Acte est réutilisé dans différentes situations (Rhodes, 1995). Ce scénario est activé à chaque fois que le système juge opportun de fournir un indice, un message d'encouragement ou d'avertissement. En regardant ce scénario de plus près (voir Figure 5.3 : Scénario pour donner un indice) on voit par exemple, pour que l'Acte « Advetize need to compose Hint » puisse s'exécuter, ses trois préconditions doivent être vérifiées et donc à Vrai. Ainsi, le système doit avoir au préalable (1) identifié la cause probable du problème rencontré par l'apprenant (préconditions : « Cause : approved »), (2) avoir sélectionné un indice à fournir par le Module du domaine (préconditions : « Hint available ») et finalement (3) la décision d'intervenir (préconditions : « Intervening approved »). Dans le cas où ces trois conditions sont vérifiées, l'affichage de l'indice se réalise.

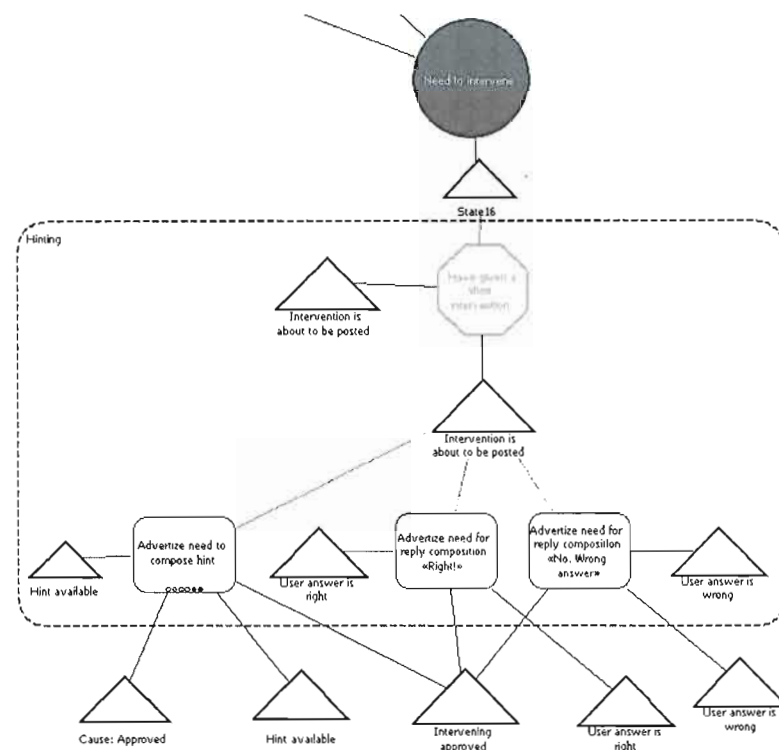


Figure 5.3 : Scénario pour donner un indice

Cinquième scénario : Le dernier scénario, qui temporellement est le premier à se produire, est responsable de la phase d'amorçage du système. En effet, dès l'exécution de CTS et avant même que les informations proviennent de la Perception, un microprocessus d'information est automatiquement créé. Il vient lancer l'Acte d'amorçage dans le réseau. Cette étape est cruciale, car elle permet de lancer une panoplie de microprocessus de raisonnement indispensable au bon fonctionnement du système. Une description du fonctionnement de certains de ces microprocessus est faite dans la section suivante.

5.4 Les microprocessus de raisonnement

Certains Actes du réseau, dont l'Acte d'amorçage, lancent des microprocessus de raisonnement qui existent tout au long de la vie du système. Ils supervisent des informations particulières qui transitent dans tout le Théâtre. Voici à titre informatif à quoi peuvent servir ces opérateurs, dont je ne présente que les plus intéressants, car il serait inutile d'énumérer les trente-trois (33) composants actuellement l'architecture.

MessageComposition : c'est un microprocessus de raisonnement qui joue le rôle de tampon de sortie. Il lit tous les microprocessus d'informations publiés par la Conscience d'accès et reconnaît ceux servant à l'affichage d'une fenêtre de communication. Il collecte le ou les champs suivants : le texte de la question ou du message à afficher, les propositions possibles liées à une question, la solution de la question et la durée d'affichage de la fenêtre. Lorsqu'il repère un de ces éléments dans une publication de la Conscience, ce microprocessus les stocke et décide de la forme finale de la fenêtre à afficher, de la manière dont les informations y sont introduites.

LearnerProfile et LearnerKnowledgeModel : ces deux microprocessus de raisonnement simulent respectivement des informations relatives au profil de l'apprenant et les connaissances possédées par l'apprenant. Bien que la simulation soit simpliste et ne tienne pas compte de la progression des apprenants, elle rend néanmoins possible de démontrer la plausibilité de toute l'architecture CTS.

TimeCounter : ce processus sert à détecter l'inactivité d'un utilisateur. Au-delà d'un temps déterminé, en l'absence d'une information spécifique auquel s'attend ce chronomètre, il crée une information qui informe de la passivité de l'apprenant et de cette manière incite CTS à passer à l'action. Ce processus peut s'utiliser dans plusieurs situations, notamment lorsque l'apprenant met beaucoup de temps pour répondre à une question ou lorsqu'il met trop de temps pour terminer un exercice. Le compteur temporel peut être activé ou désactivé à souhait par le RA et peut s'utiliser pour plusieurs activités de contrôle en même temps.

Deliberation : ce processus constitue sans doute l'un des microprocessus les plus intéressants de l'architecture. Une délibération est un processus qui consiste à examiner par la collaboration de plusieurs ressources (en l'occurrence des modules) les différents aspects d'un problème en vue de prendre une décision. À l'aide de ce microprocessus, tous les modules de l'architecture contribuent dans l'élaboration d'une réponse pour une situation problématique donnée et pour en déterminer la plus probable. Je reviendrai en détail sur la délibération dans le chapitre Six.

ExpectationCodelet : tirés de la théorie de Baars telle qu'implémentée par le professeur Franklin et son équipe, ces microprocessus de confirmation sont une variante des microprocessus de raisonnement et trouvent leur utilité dans la détection d'informations en Mémoire de travail. Ces microprocessus s'intéressent directement à l'information présente dans la Mémoire de travail et vérifient que des résultats attendus se réalisent (Homeyer, 2006). Afin d'optimiser le fonctionnement de ces processus par rapport à leur implémentation première, j'ai fait en sorte qu'ils ne détectent que les nouvelles informations dans la Mémoire de travail, ce qui permet de sauver un temps de calcul important.

5.5 Agrégation de l'information

Dans la section 5.2 précédemment décrite, j'ai énuméré les limitations du mécanisme de coalition pour lequel les agrégations d'un seul niveau sont possibles. Cette propriété empêche la formation de coalitions riches ce qui pose problème. Pour résoudre cette limitation, je décris en premier lieu la manière dont se forment les coalitions telles qu'implémentées initialement dans le système. Par la suite, je fais état de mon apport pour résoudre le problème.

Le gestionnaire de coalition (coalitionManager) qui est l'un des mécanismes de l'Attention est celui qui crée les coalitions dans la Mémoire de travail. À chaque cycle, il calcule pour

chacun des microprocessus se trouvant dans la Mémoire de travail la moyenne des « enfants » de premier niveau additionnée à la valeur dudit microprocessus (voir pseudo-code suivant). La coalition qui obtient ainsi la valeur la plus importante est sélectionnée par l'Attention, pour former la coalition « gagnante », c'est-à-dire celle qui sera publiée à tous les modules de l'architecture (incluant les microprocessus). Finalement, tous les microprocessus qui forment la coalition gagnante sont supprimés de la Mémoire de travail pour être copiés dans la zone Conscience d'accès, simulant ainsi la migration de l'information.

```

FONCTION Créer_Coalition_Gagnante
microprocessus = APPELER Retourner_Microprocessus_gagnant()
coalition_gagnante = microprocessus
    POUR chaque fil de microprocessus FAIRE
        Ajouter à la liste coalition le microprocessus fil
        Effacer fil de la mémoire centrale
    FIN POUR
    effacer de la mémoire centrale microprocessus
    retourner coalition_gagnante
FIN FONCTION

```

```

FUNCTION Retourner_Microprocessus_gagnant()
max_somme_coalition = 0 ;
max_microprocessus = null ;
POUR chaque microprocessus présent dans la Mémoire de travail FAIRE
    somme_valeur_fils = 0 ;
    nombre_de_fils = 0 ;
    somme_coalition = 0
    POUR chaque fils de premier niveau du microprocessus FAIRE
        somme_valeur_fils = somme_valeur_fils + valeur_du_fils ;
        nombre_de_fils = nombre_de_fils + 1 ;
    FIN POUR
    SI Nombre_de_fils > 0 Alors
        somme_coalition = (somme_valeur_fils / nombre_de_fils) + valeur_du_microprocessus
    SINON
        somme_coalition = valeur_du_microprocessus
    FIN SI
    SI (somme_coalition > max_somme_coalition) Alors
        max_somme_coalition = somme_coalition
        max_microprocessus = microprocessus ; //représente le microprocessus central a la
        //coalition
    FIN SI
FIN POUR
Retourne max_microprocessus
FIN FONCTION

```

Cette manière de faire reste trop limitative et ne tient pas en compte de la structure même du réseau. En effet, en regardant de plus près comment sont connectés les microprocessus d'information dans la Mémoire de travail, on s'aperçoit que tous les microprocessus ne sont pas égaux en importance. Dans la Mémoire de travail, deux catégories de microprocessus ressortent :

1. Les microprocessus d'information qui sont en quelque sorte au sommet de la hiérarchie informationnelle; je les appelle *microprocessus d'information centraux*. Ces microprocessus centraux peuvent être explicitement définis comme tels par un concepteur ou automatiquement détectés par CTS¹⁵.

¹⁵ CTS considère que tous les nœuds qui n'ont pas de liens *parent* sont des nœuds centraux.

2. Les microprocessus d'information qui sont liés au moins à un microprocessus Parent.

Il s'avère que la première famille décrit la nature de l'information, tandis que la deuxième renferme les détails de ladite information.

En reprenant, l'exemple du risque de collision, on s'aperçoit que le microprocessus central « Collision : possible » se trouvant au sommet de la hiérarchie des microprocessus décrit le type de situation, tandis que tous ses nœuds enfants renferment les détails relatifs à l'état de proximité. Dans cet exemple, la coalition pertinente à former serait celle composée du nœud central « collision : possible » ainsi que tous ses fils. CTS évite d'éventuelle boucle en interdisant la redondance des microprocessus d'information lors de la création de coalitions.

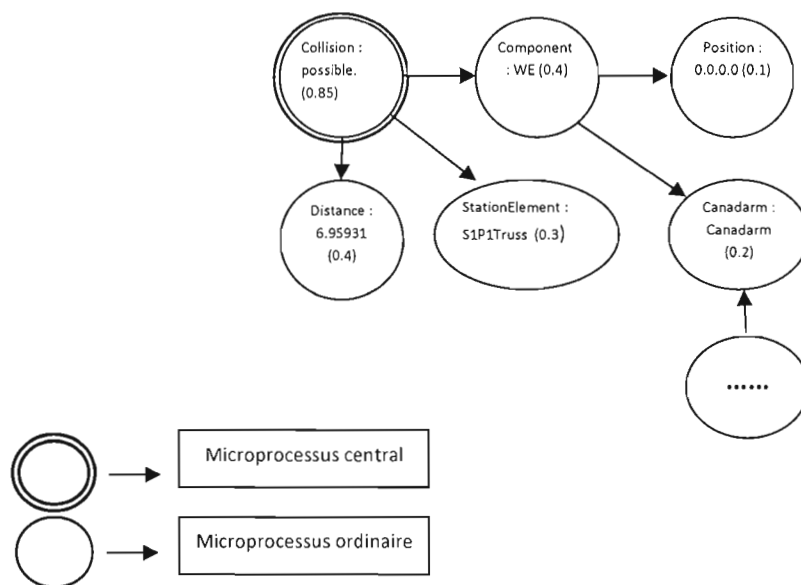


Figure 5.4 : Structure du réseau d'information dans la Mémoire de Travail

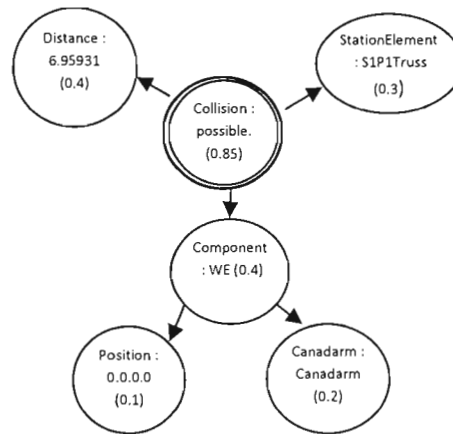


Figure 5.5 : Coalition à former

Au regard de ces descriptions, il devient important pour le **coalitionManager** de tenir compte, lors du calcul de la coalition gagnante, du nouveau paramètre qu'est le microprocesseur central. L'introduction de ce nouveau critère, ainsi que la possibilité de créer des coalitions avec différents niveaux de profondeurs, oblige la mise en place d'une équation plus riche qui se doit de respecter les critères suivants :

1. Une coalition formée par un microprocesseur central est favorisée par rapport à une coalition sans nœud central.
2. Un réseau de microprocesseurs d'information dans lequel vient se greffer ou se dissocier une nouvelle information voit son importance augmenter ou décroître.
3. L'ajout d'une information modifie la valeur de toute la hiérarchie composant le réseau.

Ces trois paramètres trouvent réponse dans l'équation suivante :

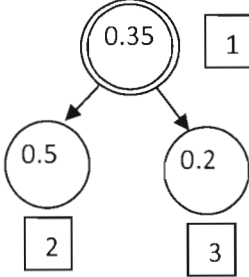
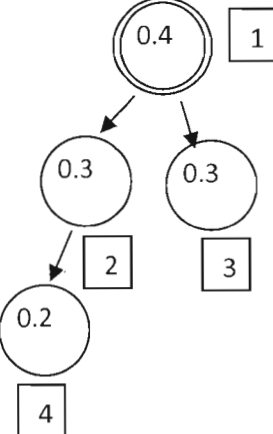
1. Posons comme *microprocessus central* un nœud spécifié préalablement par le concepteur ou qui n'a pas de liens parents.
2. Soit R , tous les microprocessus d'information présent dans la mémoire de travail.
3. Soit A , des microprocessus connectés dans la Mémoire de travail et C l'ensemble des *microprocessus centraux*, tels que $C \subseteq A$ et $A \subseteq R$
4. Soit $O = A - C$, les microprocessus ordinaires.
5. Soit $a_i \in A$, représente le i -ème microprocessus dans la Mémoire de travail.
6. Soit la fonction $F(a_i)$ qui retourne l'ensemble des fils de premier niveau de l' i ème microprocessus, tel que $F(a_i) \subseteq C$.
7. Soit la fonction $G(a_i)$ qui retourne l'ensemble des fils de premier niveau de l' i ème microprocessus, tel que $G(a_i) \subseteq O$.
8. Soit la fonction $N(a_i^t)$ qui retourne l'ensemble des nouveaux fils ajoutés à a_i à l'instant t .
9. Soit la fonction $S(a_i^t)$ qui retourne l'ensemble des nouveaux fils supprimés à a_i à l'instant t .
10. Soit nbr , un réel tel que $nbr \approx 0,2$ (voir figure 7.2).

$a_i.val += \sum F(a_i).valeur + (\sum G(a_i).valeur / \text{card}(G(a_i))) + nbr * N(a_i) - nbr * S(a_i)$ pour $\text{card}(G(a_i))$ différent de zéro.

$a_i.val += \sum F(a_i).valeur + nbr * N(a_i) - nbr * S(a_i)$ pour $\text{card}(G(a_i))$ égal à zéro.

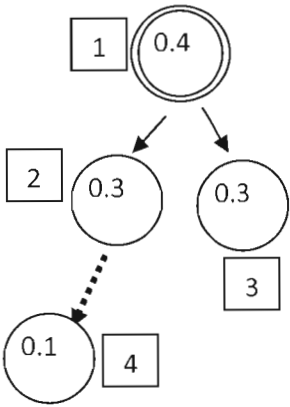
Le tableau 5.4 illustre des exemples concrets de la nouvelle équation pour la création de la coalition.

Tableau 5.4 : Récapitulatif des interactions

Microprocessus présents dans la Mémoire de travail	Description	Calcul	Conséquence
	Il existe un seul microprocessus de type central.	$a3.val=0.2$ $a2.val=0.5$ $a1.val=0.35+(0.5+0.2)/2=0.7$	Les microprocessus a1, a2 et a3 sont choisis pour former la coalition gagnante car la coalition ayant a1 à sa tête renferme la valeur la plus grande.
	Il existe un seul microprocessus de type central.	$a4.val=0.2$ $a3.val=0.3$ $a2.val=0.3+0.2/1$ $a2.val=0.5$ $a1.val=0.4+(a2.val+0.3)/2=0.8$	Les microprocessus a1, a2, a3 et a4 sont choisis pour former la coalition gagnante car la coalition ayant a1 à sa tête renferme la valeur la plus grande.

Le calcul de la coalition tient aussi compte du fait qu'un microprocessus d'information qui vient s'associer ou de se dissocier d'un autre microprocessus d'information a un impact sur la valeur d'activation de l'ensemble de la hiérarchie composant le réseau de microprocessus (voir exemple suivant). La valeur d'activation de la nouvelle information se répercute en cascade sur l'ensemble de la structure qui la chapeaute.

Tableau 5.5 : Récapitulatif des interactions : ajout d'une nouvelle information.

Microprocessus présent dans la Mémoire de Travail	Description	Calcul
	Supposons que le microprocessus a4 vient d'être ajouté à a2. Cet ajout change consécutivement la valeur de a2 et de a1.	$a2' = 0.3 + (0.1/1) + \underline{0.2}^* = 0.6$ $a1' = 0.4 + (0.6 + 0.3)/2 = 0.85$ La coalition gagnante est formée de a1, a2, a3 et a4. *La valeur nbr = <u>0.2</u> est additionnée à chaque établissement de liens dans la coalition.

5.6 Description du Réseau des Actes (RA)

L'élément le plus fragile et le plus vulnérable de l'architecture est le RA. En effet, tout au long du développement de CTS, l'essentiel des difficultés provenait des équations qui gouvernent l'écoulement de l'énergie dans le Réseau des Actes (Atkinson-Abutridy et Carrasco-Leon, 1999 ; Rhodes, 1995 ; Tyrell, 1994 ; Dorer, 1999). Il était donc important de dépasser ces limites et de trouver de nouvelles équations.

Pour ce faire, j'expose en premier lieu le fonctionnement originel du réseau tel que décrit par Maes (1989). Dans un second lieu, j'expose en détail ses limites. Par la suite je présente les solutions que j'ai apportées.

Le réseau de Maes se définit comme la combinaison d'idées émanant de l'IA (calcul distribué), la vie artificielle (propriétés émergentes) et de l'ethnologie (motivations) (Tyrell, 1994). C'est un réseau distribué et non hiérarchisé de noeuds. Tout le réseau est parcouru par une énergie (des valeurs numériques) qui provient de deux sources possibles. En

premier lieu, il y a les États qui sont des capteurs à l'écoute de certains changements de l'environnement. Ils sont initialement à faux et, dans ce cas, ils ne transmettent de l'énergie que du **haut vers le bas**. Ils deviennent vrais dès l'instant où ils reconnaissent une information particulière de l'environnement (exemple, l'utilisateur a changé une des caméras du panneau de contrôle). Dans ce cas, L'État transmet de l'énergie uniquement du **bas vers le haut**. Deuxièmement, il existe les *Désirs* (Maes les nommait « Goals ») qui constituent des objectifs globaux au système. Ils s'activent (versent de l'énergie du haut vers le bas) et se désactivent (ne versent aucune énergie) dans certaines conditions. Un exemple de *Désir*, chez CTS, tient dans le souhait d'entamer une discussion avec l'utilisateur.

Par ailleurs, chaque Acte ou Désir du réseau est connecté à un autre par l'entremise d'au moins un État. Les Actes se connectent aux États à l'aide de trois types de liens possibles : **précédence**, **succession** et **inhibition** (pour plus de détails voir section 4.2). Chacun de ces liens est régi par une équation qui définit comment l'énergie est transmise d'un nœuds à un ou plusieurs autres. L'activation provenant des États ou des *Désirs* est véhiculée dans tout le réseau à travers les trois types de connexions précédemment citées. Je transcris ci-dessous les principales équations qui régissent l'écoulement de l'énergie du réseau (je fais abstraction des équations d'inhibition et de normalisation à des fin de clareté) :

➤ **Équation 1 :** propagation de l'énergie du **bas vers le haut**

$$spreads_fw(x, y, t) = \begin{cases} \sum_j \alpha_x(t-1) \frac{\theta}{\gamma} \frac{1}{\#M(j)} \frac{1}{\#c_y} & \text{if } executable(x, t) = 1 \\ 0 & \text{if } executable(x, t) = 0 \end{cases}$$

L'envoi de l'énergie d'un Acte **X** contenant une quantité d'énergie = α_x , vers un autre Acte **Y** contenant une énergie = α_y dans le sens **avant** (du bas vers le haut) se fait uniquement quand l'État qui unit X et Y est faux et que l'Acte X est exécutable. L'équation 1 décrit combien l'Acte Y va recevoir d'énergie de l'Acte X. Dans notre cas θ/γ représente un valeur

comprise entre $[0,1]$ paramétrée par le concepteur.

j : est un indice parcourant l'ensemble des États à faux qui lient X à Y .

$\#M(j)$: le nombre de liens de **succession** de l'État j

$\#c_y$: le nombre de liens de **précédence** de l'Acte Y .

t : le temps à l'instant (t) .

➤ **Équation 2** : propagation de l'énergie **du haut vers le bas**

$$spreads_bw(x, y, t) = \begin{cases} \sum_j \alpha_x(t-1) \frac{1}{\#A(j)} \frac{1}{\#a_y} & \text{if } executable(x, t) = 0 \\ 0 & \text{if } executable(x, t) = 1 \end{cases}$$

Cette équation est presque identique à l'équation 1. Pour qu'il y ait une propagation vers l'arrière (du haut vers le bas), il faut que l'Acte ne soit pas exécutable. Je suppose dans ce cas que l'énergie se propage du haut vers le bas, de l'Acte X vers l'Acte Y . Cette équation décrit combien l'Acte Y va recevoir d'énergie de l'Acte X .

J : est un indice parcourant tous les États à faux qui lient X à Y .

$\#A(j)$: le nombre de liens de **précédence** de l'État j .

$\#a_y$: le nombre de liens de **succession** de l'Acte Y .

➤ **Équation 3** : entrée de l'énergie à partir des Désirs **du haut vers le bas**

$$input_from_goals(x, t) = \sum_j \gamma \frac{1}{\#A(j)} \frac{1}{\#a_x}$$

Cette équation décrit comment l'énergie est insufflée par les Désirs. Le mouvement de l'énergie se fait de haut en bas, du Désir D vers l'Acte X . γ représente l'énergie insufflée par le Désir D . γ est une valeur déterminée par le concepteur et elle est comprise entre $[1,100]$.

J : est un indice qui parcourt les États faux qui lient D à X .

#A(j) : le nombre de liens de **précédence** de l'État j.

#a_x : le nombre de liens de **succession** de l'Acte X.

➤ **Équation 4** : Entrée de l'énergie à partir des États **du bas vers le haut**

$$input_from_state(x, t) = \sum_j \phi \frac{1}{\#M(j)} \frac{1}{\#c_x}$$

Cette équation décrit comment l'énergie est insufflée par les État vers un Acte X. Ici on suppose que l'énergie se propage d'un État donné vers un Acte X. Le symbole ϕ représente la quantité d'énergie insufflée par l'État Vrai. Cette quantité est comprise entre [1, 100] et est spécifiée par le concepteur.

J représente tous les États vrais qui sont liés par un lien de **succession** avec l'Acte X.

#M(j) : le nombre de liens de **succession** de l'État j.

#c_x : le nombre de liens de **précédence** de l'Acte X.

➤ **Équation 5** : Équation régissant la valeur énergétique actuelle de l'Acte

$$\begin{aligned} \alpha(y, 0) &= 0 \\ \alpha(y, t) &= \alpha(y, t-1) \\ &+ \sum_{x,z} (spreads_bw(x, y, t) + spreads_fw(x, y, t) - takes_away(z, y, t)) \\ &+ input_from_state(y, t) + input_from_goals(y, t) \end{aligned}$$

L'équation 5 décrit l'énergie accumulée par un Acte en fonction du temps. Toutefois, il faut ajouter deux autres équations dont j'ai volontairement fait abstraction afin de ne pas alourdir la compréhension de la mouveance de l'énergie dans le réseau. Ces deux équations

concernent¹⁶ :

1. L'inhibition (**takes_away**): elle permet de diminuer l'énergie d'un Acte dans le cas où il y a un lien d'inhibition.
2. La normalisation : elle ajuste l'activation globale du réseau afin de la maintenir à un seuil moyen déterminé. Elle s'exécute à chaque cycle cognitif.

La figure suivante reprend les équations numéro 2 (gauche) et 4 (droite) afin d'exposer à travers un exemple graphique la mouvance de l'énergie dans le réseau. Y représente la quantité d'énergie introduite par le Désir et O représente celle introduite par l'État vrai.

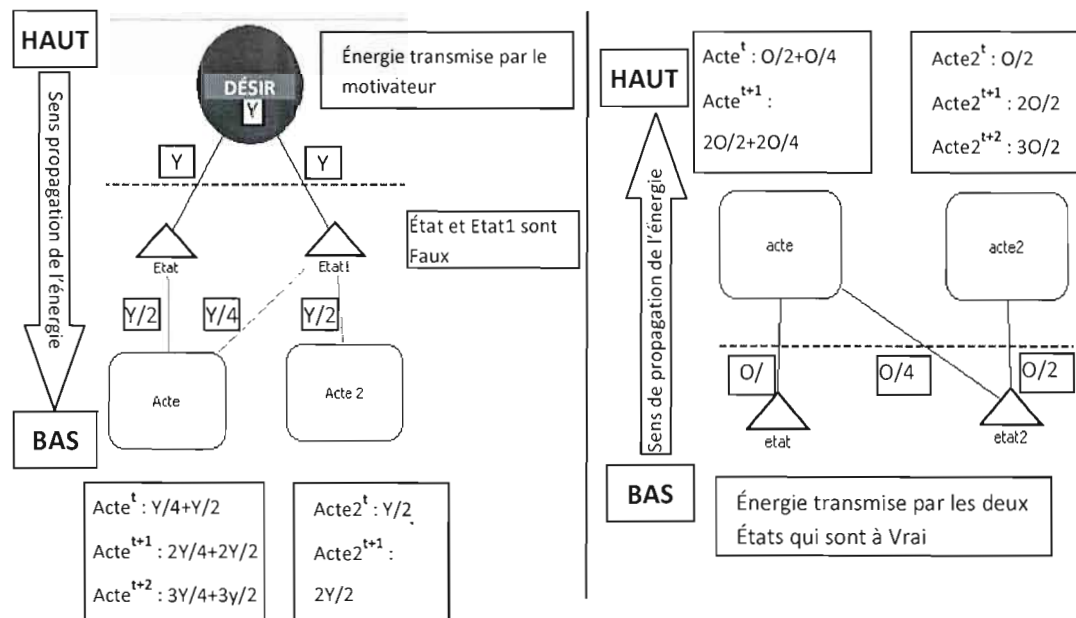


Figure 5.6 : Dispersion de l'énergie dans le RA

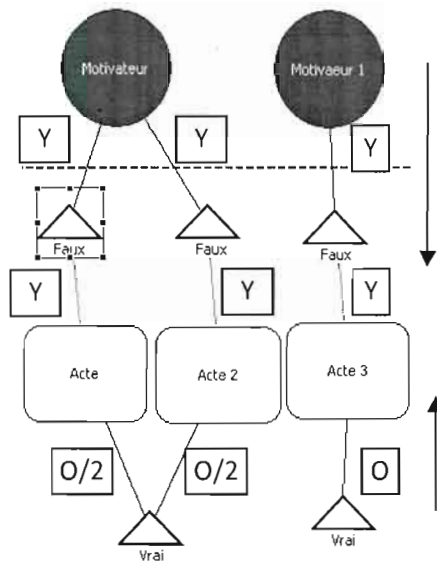
¹⁶ Pour une description plus détaillée du fonctionnement du réseau, veuillez vous reporter à l'excellent article de Maes (1989) qui décrit mathématiquement la manière dont s'écoule l'énergie dans le réseau.

En fait, l'idée de base du RA est qu'en fonction de l'écoulement de l'énergie qui varie selon les besoins internes de l'agent (provenant des Désirs), des excitateurs externes de l'environnement (représentés par des États) et des liens d'inhibitions, certains nœuds se trouvent plus activés (ont plus d'énergie) que d'autres et obtiennent priorité pour s'exécuter. Ces Actes ne s'enclenchent que lorsque toutes leurs préconditions sont vraies.

5.7 Modification du fonctionnement du Réseau des Actes

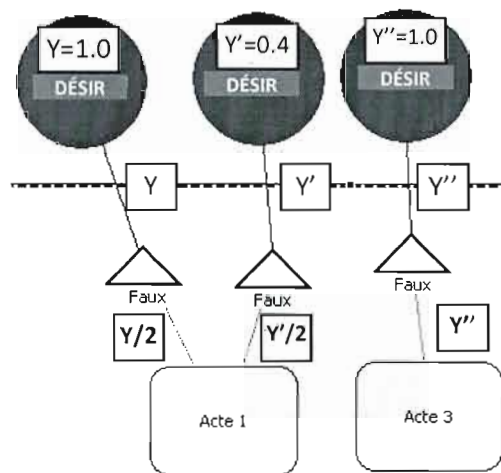
Tyrell (1994) a analysé de plus près le réseau de Maes et y a décelé différentes limitations. Nous avons aussi rencontré ces limitations lors de l'implémentation du Réseau des Actes de CTS. J'en donne ici un compte-rendu exhaustif :

Première limitation du RA citée par Tyrell :



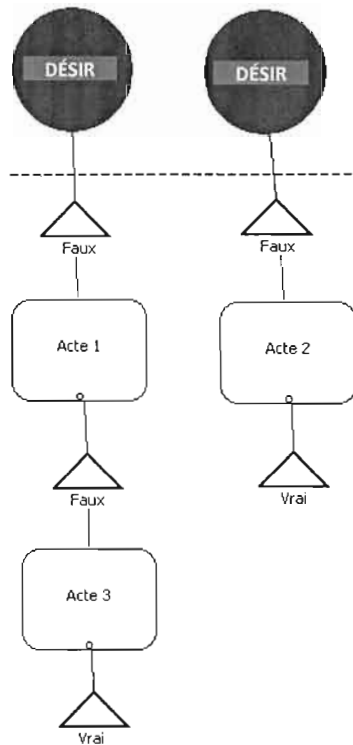
Dans l'exemple, l'énergie (Y) coule du haut vers le bas via les Désirs 1 et 2. L'énergie (O) coule aussi du bas vers le haut via les deux États qui sont à Vrai. Au regard de la mouvance de l'énergie, l'Acte numéro 3 sera plus énergétique que les deux autres. Cet Acte est amené à s'exécuter avant les deux autres. Pourtant, les Actes numéro 1 et 2 peuvent être aussi importants que l'Acte 3 et ne devraient pas se voir désavantagés par la structure du réseau.

- Deuxième limitation du RA citée par Tyrell :



Dans cet exemple, l'énergie (Y) coule du haut vers le bas uniquement. Tous les États s'affichent actuellement comme non satisfaits, c'est-à-dire à faux. Dans cet exemple, l'Acte 1 aura accumulé une énergie de valeur $0.7 = 1.0/2 + 0.4/2$. Cependant l'Acte 3 aura comme énergie la valeur de 1. Or, dans ce cas de figure, il est préférable d'exécuter l'Acte 1 plutôt que l'Acte 3, car il permet de satisfaire deux Désirs.

- Troisième limitation du RA citée par Tyrell :



Dans cette configuration, l'énergie (Y) coule dans les deux sens. Il s'avère que l'Acte 3 accumule beaucoup plus d'énergie que l'Acte 2 et l'empêche de s'exécuter. Or, dans ce cas de figure, il est préférable d'exécuter l'Acte 2 que l'Acte 3, car il permet de satisfaire instantanément le but. Le graphe suivant expose l'évolution de l'énergie au sein des Actes. On remarque qu'à partir de T=3 l'énergie de l'Acte 3 est supérieure à l'acte 2.

	T=0	T=1	T=2	T=3
Acte 1	Zéro	$Y=(1)$	$2Y+O^2/Y$ $= (4)$	$3Y+3$ $O^2/Y+Y$
Acte 2	Zéro	$O+Y=(2)$	$2O+2Y=(5)$	$3O+3Y$
Acte 3	Zéro	$O=(3)$	$2O+Y=(6)$	$3O+3Y+$ O^2/Y

Explication :

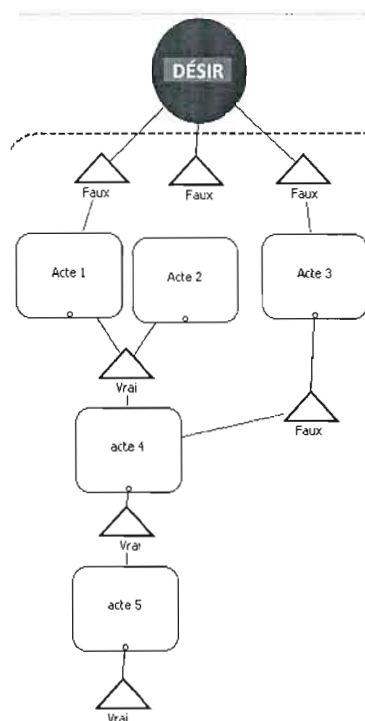
$$(4^*) = (1) + O/Y * (3) + Y = Y + O/Y * O + Y = 2Y + O^2/Y$$

$$(5) = (2) + O + Y = 2O + 2Y$$

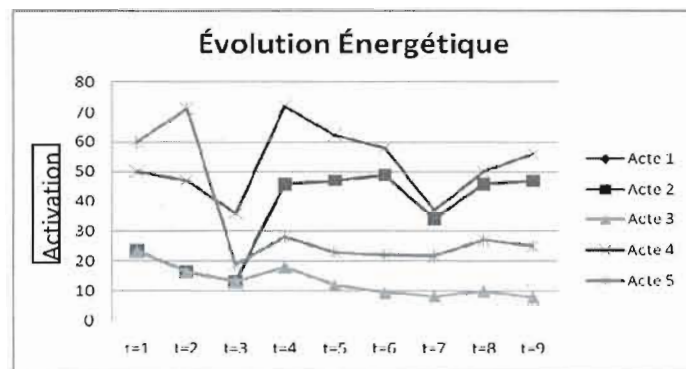
$$(6) = (3) + Y + O = 2O + Y$$

* L'activation de l'acte1 (t=2) est la somme de l'énergie de l'acte1 (t=1) additionné à une fraction de l'énergie de l'acte3 (t=1) additionné à l'énergie provenant de l'état Vrai (Y).

Par ailleurs, une autre limitation de RA non décelée par Tyrell est le fait que Maes n'a pas tenu compte des États qui sont toujours vrais. Or, le fait de garder des États à Vrai pose problème et **bloque** le réseau dans certaines situations. Le schéma suivant expose graphiquement le problème :



Dans le cas de ce réseau, il se produit à l'instant $T=2$ l'exécution de l'Acte 5. Dès l'instant où $T \geq 3$, l'Acte 4 contient beaucoup plus d'énergie que les Actes 1 ou 2 (sur le graphique, les deux courbes sont superposées). L'Acte 4 les empêche de s'exécuter car il contient le plus d'énergie du réseau. Cela résulte du fait que l'Acte 4 accumule plus d'énergie qui provient à la fois de l'Acte 5 et de l'acte 3.



Toutes ces comportements non souhaitables font en sorte que le fonctionnement du réseau reste fragile, parfois même inapproprié. Ce constat est d'autant plus vrai que lors du développement du RA dans CTS, plusieurs problèmes précédemment cités ont vu le jour et ont rendu le développement du Réseau des Actes complexe, voire impossible. Il était ainsi nécessaire d'apporter les rectifications nécessaires afin de garantir un fonctionnement cohérent de l'agent CTS. Pour ce faire, j'ai apporté les modifications suivantes au RA.

J'ai introduit des liens de type ET et OU entre les États et les Actes. Cet ajout permet d'enrichir le fonctionnement du réseau et de changer la manière dont s'écoule l'énergie. Ces nouveaux liens rajoutent un niveau de souplesse supplémentaire qui n'existait pas dans les précédents travaux relatifs au Réseau des Actes.

Un lien de type OU (ou exclusif) est utile dans le cas où, pour un groupe d'États connectés à un Acte donné, un seul d'entre eux peut être vrai. Les liens OU reflètent des choix potentiellement vrais, mais dont un seul est possible à un instant donné. Un exemple concret serait quand l'utilisateur répond juste ou incorrect à une question donnée. Sa réponse vient mettre à Vrai l'un des deux États uniquement (voir Figure 5.7 : Exemple de liens OU.). Il suffit **qu'un** seul de ces États soit mis à Vrai pour que les autres États appartenant au groupe deviennent **caduques et sont comme supprimés du réseau**. Dans l'exemple suivant, il suffit que l'un des deux États soit à Vrai pour que le deuxième cesse d'envoyer de l'énergie vers l'arrière, c'est-à-dire vers l'Acte « Wait for user answer to Q-IR-7 ».

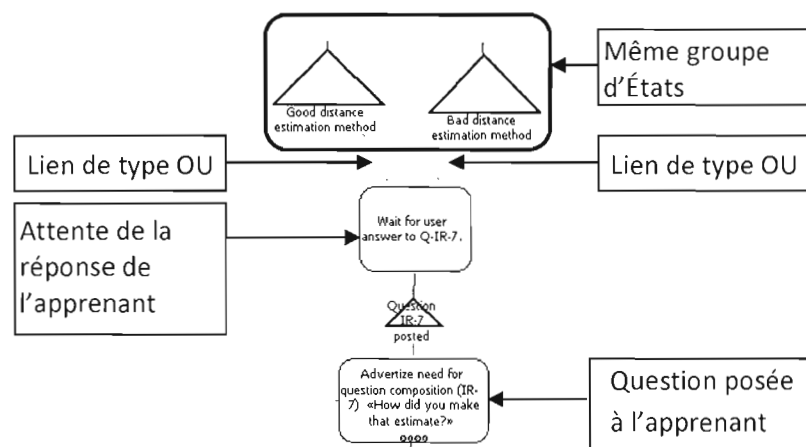


Figure 5.7 : Exemple de liens OU.

À l'opposé un lien de type ET est utile dans le cas où tous les États d'un groupe doivent être vérifiés et donc vrais. Les États appartenant à un même groupe traduisent un ensemble de conditions complémentaires et indispensables dans la réalisation d'un Acte. Un lien de type ET reflète des choix dépendant les uns des autres et qui doivent être vérifiés à un instant donné pour permettre le déclenchement d'un Acte. Pour illustrer ces affirmations, je suppose que l'agent CTS souhaite intervenir lors d'une mauvaise manipulation; pour permettre la mise en œuvre de ce désir, les États « User answer is wrong » et « Intervening approved » doivent être les deux à vrai (voir exemple de la Figure 5.8 : Exemple de liens ET).

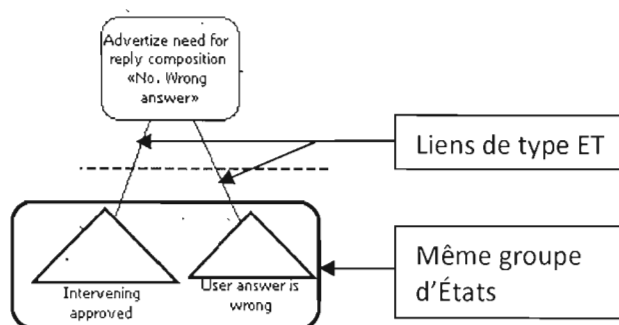


Figure 5.8 : Exemple de liens ET

Voici une description mathématique de la circulation de l'énergie concernant toutes les modifications réalisées :

L'équation suivante vient remplacer l'équation numéro de Maes. Elle décrit comment s'opère l'entrée de l'énergie à partir du nœud Désir D vers un acte X

$$\left\{ \begin{array}{ll} \sum_j \gamma^* (1/\#A(j) * \#a_y) & \text{pour } A(j) \in E \\ \text{Sinon} & \\ \sum_j \gamma^* & \left\{ \begin{array}{ll} 1 & \text{si } \#(a_y)=0 \text{ pour } a_y \in E \text{ et } A(j) \in O \\ 1/\#(a_y) & \text{avec } a_y \in E \text{ et } A(j) \in O \end{array} \right. \end{array} \right.$$

J représente les États qui unissent D à X.

A(j) : représente les liens de précédence de l'État j;

a_y est l'ensemble des liens de succession de l'Acte X.

O représente l'ensemble des liens de type OU qui forment le RA.

E représente l'ensemble des liens de type ET qui forment le RA.

symbolise la cardinalité.

L'équation suivante vient remplacer l'équation numéro de **2** de Maes. Elle dicte comment l'énergie circule de haut en bas, de l'Acte X vers l'Acte Y pendant le temps (t).

$$\left\{ \begin{array}{ll} \sum_j a_x(t-1) * (1/\#A(j) * \#a_y) & \text{pour } A(j) \in E \\ \text{Sinon} & \\ \sum_j a_x(t-1) * & \left\{ \begin{array}{ll} 1 & \text{si } \#(a_y)=0 \text{ pour } a_y \in E \text{ et } A(j) \in O \\ 1/\#(a_y) & \text{avec } a_y \in E \text{ et } A(j) \in O \end{array} \right. \end{array} \right.$$

J représente les États qui unissent l'Acte X et Y.

A(j) : représente les liens de précédence de l'État j;

a_y est l'ensemble des liens de succession de l'Acte Y.

O représente l'ensemble des liens de type OU.

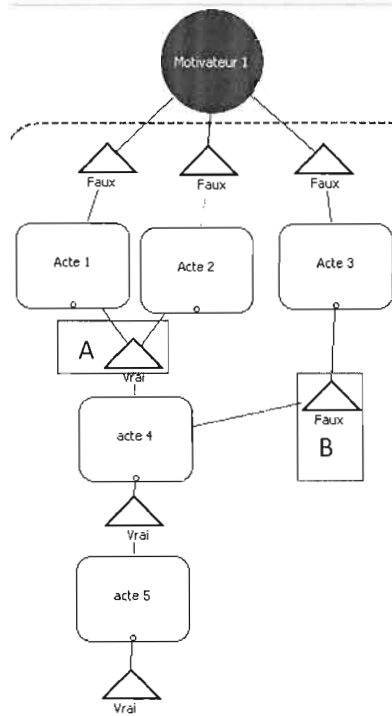
E représente l'ensemble des liens de type ET.

symbolise la cardinalité.

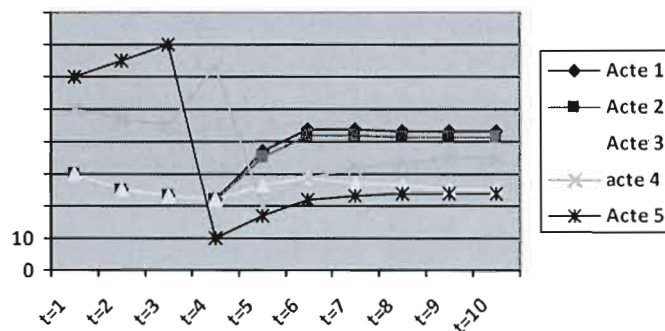
De même, il est possible, en s'appuyant sur l'équation ci-dessus, d'extraire par analogie une nouvelle équation qui régit la mouvance de l'énergie du bas vers le haut. Il suffit simplement de remplacer a_y par c_x , $A(j)$ par $M(j)$ et introduire le facteur θ/γ .

Avec ces nouvelles équations, les limites citées par Tyrell (1994) sont en partie éliminées.

- En reprenant les limites précédemment citées, il s'avère que pour la première limite citée par Tyrell, il suffit de modifier les deux liens de précédence des deux actes 'Acte' et 'Acte2' de type OU. Ceci aura comme conséquence que les trois Actes prendront la valeur 1.
- Dans l'exemple 2, l'Acte 1 est appelé à avoir des liens de successions de type OU. Ainsi l'Acte 1 aura une valeur de 1.4 et sera exécuté en priorité.
- Pour l'exemple 3, il est possible de contourner le problème en posant $Y > O$ et le système réagit normalement.
- Finalement, pour la dernière limitation, en introduisant des liens de types OU pour les liens de succession de l'Acte 4, le réseau se remet à fonctionner correctement. L'exemple suivant démontre le nouveau fonctionnement du réseau.



La courbe-ci dessous dévoile un comportement nouveau et conforme aux attentes. Les Actes 1 et 2 sont les plus énergétiques (à partir du temps $t > 6$) et se déclenchent sans problème. Cela résulte du fait que l'Acte 4 dispose de deux liens de type OU avec les États A et B. Ainsi, dès l'instant où l'État A devient vrai, l'État B bloque l'envoi de l'énergie vers l'arrière ce qui permet aux Acte 1 et 2 d'accumuler plus d'énergie que l'Acte 4 et ainsi de s'exécuter normalement.



La nouvelle approche apporte au réseau un fonctionnement plus fin et facilite la mise en place d'un réseau étendu. Avec cette nouvelle manière de faire, il a été possible de passer d'un scénario comprenant une dizaine d'Actes et au fonctionnement inapproprié à un réseau composé d'une trentaine d'Actes et fonctionnant correctement. Sans de telles modifications, il serait impossible de développer un réseau aussi riche et étendu.

CHAPITRE VI : ÉMOTION LORS DU TUTORAT

Je viens d'exposer dans les précédents chapitres, les réalisations et implémentations apportées au système afin de garantir un fonctionnement meilleur et adéquat de CTS (voir **Figure 6.1** : Les ajouts faits à CTS) Dans cette partie, je propose une extension théorique (Gaha, Dubois et Nkambou, 2008). L'architecture de CTS est suffisamment riche pour autoriser plusieurs ajouts. J'ai donc essayé à ma manière d'apporter un aspect nouveau au système CTS, à savoir les émotions.

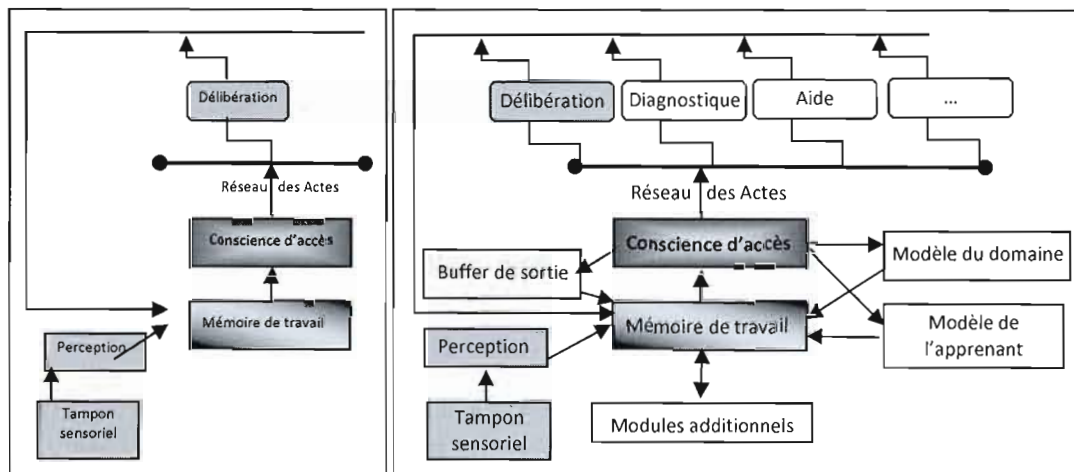


Figure 6.1 : Les ajouts faits à CTS. Le graphe de gauche représente les modules implémentés à l'origine dans CTS. À droite, on voit les ajouts que j'ai faits pour rendre CTS fonctionnel et le doter de réactions pédagogiques. Par « *Modules additionnels* », je fais référence aux travaux de Usef Faghihi qui concerne le volet apprentissage (Faghihi, Dubois et Nkambou, 2007).

6.1 Le pourquoi de l'émotion (Gaha, Dubois et Nkambou, 2008)

Pour les théoriciens interactionnistes, le succès de tout processus d'apprentissage dépend grandement des prédispositions émotionnelles de l'apprenant. En effet, selon eux, l'intelligence dépend étroitement de l'émotion. Daniel Goleman (Goleman, 1995), fut l'un des premiers à attirer l'attention du grand public sur la théorie de l'intelligence émotionnelle et à soutenir qu'elle nous pilote jour après jour. Selon lui, les plus académiquement brillants d'entre nous se trouvent vulnérables aux échecs ou dysfonctionnements occasionnés par des émotions débridées. Ainsi, les émotions tiennent un rôle fondamental dans tout processus cognitif et se révèlent primordiales lors de l'apprentissage. Parce qu'elles interagissent avec les sentiments et les états psychologiques et biologiques de l'individu, les émotions, comme le soutiennent Alvarado et ses collègues (Alvarado, Adams et Burbeck, 2002), agissent sur l'attention de l'apprenant et sur ses capacités de compréhension et de mémorisation. Un apprenant vivant une détresse émotionnelle (craintes d'échouer, peur, pressions psychologiques de diverses natures, sentiment d'incompétence, etc.) verra ses performances d'apprentissage réduites (Thomas *et al.*, 2001). Par contre, s'il est émotionnellement soutenu, l'apprenant présentera des résultats meilleurs et son apprentissage sera mieux réussi.

Pour toutes ces raisons, plusieurs STI prennent aujourd'hui en considération les aspects émotionnels de l'apprenant. Des systèmes tuteurs tels que STEVE (Elliot, Rickel et Lester, 1999), COSMO (Lester, Towns et Fitzgerald, 1999), PAT (Augustin *et al.*, 2004) et DARWAR (Johnson, Vilhjalmsen et Samtani, 2006) prennent appui sur des recherches en psychologie et/ou en neurologie pour dispenser des enseignements qui tiennent compte de la dimension émotionnelle. Notre agent s'inscrit dans la même perspective. Il lui manquait cependant la définition de mécanismes modélisant les états émotionnels de l'apprenant et portant effet sur les décisions de l'agent. Je propose ici une telle extension.

Le présent chapitre se fixe comme objectif premier de présenter cette extension de l'architecture de CTS rendant possible un apprentissage personnalisé et évolutif tout en tenant compte des aspects émotionnels de l'apprenant. Pour les besoins de l'analyse, j'examine dans la prochaine section le lien entre émotion et cognition. Dans la section 8.3, je présente ma proposition théorique d'extension de l'aspect émotionnel chez CTS, pour enchaîner enfin avec l'explication des interactions fonctionnelles du système.

6.2 Émotion et cognition

Historiquement, les émotions et la cognition ont souvent été considérées comme deux notions indépendantes. Pour plusieurs philosophes et auteurs classiques, les émotions sont plutôt récusables; elles affectent négativement la cognition *et altèrent* l'exercice de la raison. Platon, déjà, décrivait les émotions comme "des désirs irrationnels qui doivent être contrôlés lors de l'utilisation de la raison". Ce point de vue a dominé la pensée scientifique et s'est perpétué. On le retrouve chez Kant qui affirme que les émotions sont une "maladie du cerveau" (O'Regan, 2003). Cette acception négative des émotions est toujours présente dans la littérature. Elle a caractérisé les recherches et les théories afférentes à l'éducation jusqu'à très récemment. Selon plusieurs analystes, tels que rapportés par Ferro (Ferrero, 1995), Martinez (Martinez, 2001) les émotions et la cognition sont deux entités distinctes et pour mieux apprendre, seule la cognition compte.

Or, de récentes recherches menées en sciences cognitives et en neurosciences ont démontré que les émotions jouent un rôle central dans l'apprentissage et dans la prise de décision (Ahn et Picard, 2005) (Adam et Evrard, 2005) (Chaffar, Chalfoun et Frasson, 2006) (Ahn et Picard, 2005). Bechara et ses collègues (Bechara *et al.*, 1997) soutiennent qu'une personne raisonne, décide et agit à partir et à la lumière des expériences émotionnelles vécues. Dans la même perspective, Postle (Postle, 1993) développe un modèle multimodal où les actions à entreprendre et l'apprentissage à réaliser dépendent de l'état émotionnel de l'Acteur. En neurobiologie, LeDoux (LeDoux, 1996) démontre, à travers des tests

cliniques, la relation entre l'émotion et certains aspects de la cognition. Enfin, les travaux de Lisetti (Lisetti et Nasoz, 2004) confirment, de manière empirique, l'étroite relation entre cognition et émotion; ils recensent neuf aspects où la cognition serait directement influencée par les émotions, parmi lesquels on trouve l'apprentissage. Lisetti affirme que les gens apprennent plus ou moins correctement selon leur état émotionnel. Augustin et ses collègues (Augustin *et al.*, 2004) ajoutent que l'induction d'émotions plutôt «favorables» autorise une meilleure réalisation du processus d'apprentissage et une efficacité au niveau des objectifs cognitifs. Selon Alvarado (Alvarado, 2002), tout apprentissage efficace tient nécessairement compte de l'aspect émotionnel de l'apprenant. Un tutorat qui néglige une telle dimension prend le risque d'omettre un facteur déterminant dans le succès de l'apprentissage. Ainsi, un système tuteur capable de tenir compte de l'état affectif de l'apprenant pour construire une approche pédagogique appropriée serait à même de soutenir l'apprenant et de lui permettre de mieux réussir ses apprentissages.

Aujourd'hui, plusieurs systèmes d'apprentissage cherchent donc à inclure dans leur mode opératoire une telle dimension. Un système tuteur qui néglige une telle dimension prend le risque d'omettre un facteur déterminant dans le succès de l'apprentissage (Alvarado, 2002).

Pour optimiser le processus d'apprentissage, CTS doit être capable d'induire des émotions opportunes chez l'apprenant, des émotions de nature à l'aider à mieux apprendre. Pour ce faire, il doit opter pour une approche pédagogique/didactique appropriée qui s'adapte au mieux à l'état émotionnel de l'apprenant.

Après des recherches et concertation avec quelques membres du projet CTS il a été convenu d'intégrer l'aspect émotionnel avec un mécanisme en deux parties. En premier lieu, la portion concernant les émotions de CTS lui-même tient dans une collection d'Émotions positionnés dans le Réseau des Actes et qui s'impliquent dans le choix des actions du tuteur. En deuxième lieu, je compte décrypter l'état émotionnel de l'apprenant via à l'aide d'une variante du Modèle OCC (Ortony, Clore et Collins 1988) intégrée dans le

MA. De l'interaction entre la représentation de l'état émotionnel de l'apprenant et des Émotions représentées dans le planificateur de tâche résulte un schéma d'apprentissage conséquent et plus personnalisé. Pour mieux comprendre cette approche, j'analyse, en premier lieu, les fondements théoriques du système émotionnel. Dans un second lieu, je donne un exemple d'exécution en citant les avantages de cette approche en comparaison à CTS dans son état original, sans la gestion émotionnelle.

6.3 Description théorique

Pour tenir compte de l'aspect émotif de l'apprenant, CTS est appelé à traiter deux ensembles d'excitateurs à l'aide d'une mécanique émotionnelle qui se compose (1) des Émotions du tuteur CTS, implémentées sous forme de capteurs spéciaux qui viennent se greffer au niveau des Désirs du RA et (2) du Modèle de l'apprenant. Ces excitateurs couvrent deux champs : les excitateurs **externes** qui sont des événements émotionnels/cognitifs captés de l'apprenant (Ahn et Picard, 2006), et les excitateurs **internes** provenant des conclusions tirées par le Modèle de l'apprenant.

6.3.1 Excitateurs externes

Les excitateurs externes apparaissent habituellement comme conséquence directe de la stratégie pédagogique ou du soutien émotionnel adopté par le tuteur. Ils représentent les signes d'un état émotionnel/cognitif particulier attendu chez l'apprenant. En effet, certaines actions faites par CTS prévoient l'apparition de manifestations émotionnel/cognitif particulières chez l'apprenant (telles que le rehaussement du taux de succès, l'accélération des prises de décision de l'apprenant, état de motivation chez l'apprenant...). Dans le cas où cet exciteur externe attendu est détecté, le tuteur se reconnaît satisfait et conforte l'action qu'il a prise. Par contre, en l'absence de l'effet escompté, le tuteur est insatisfait, ce qui doit se transformer en une volonté d'essayer une autre stratégie pédagogique pour

atteindre l'état cognitif espéré. L'état d'insatisfaction du tuteur peut aussi mener à la recherche d'une nouvelle manière de faire en vue d'induire un état émotionnel plus approprié. Pour mieux expliciter cette idée, prenons le cas où le Modèle de l'apprenant amène le tuteur à supporter l'apprenant en favorisant, dans le RA, l'exécution des sous séquences d'Actes pour le motiver. Des microprocessus de confirmation sont alors lancés et seront à l'affût d'événements extérieurs spécifiques reflétant un meilleur état de motivation chez l'apprenant. Ces événements de nature émotionnelle ou cognitive représentent les excitateurs externes attendus. Ils sont des manifestations révélant que l'apprenant a été correctement soutenu. S'il y a absence de ces excitateurs externes après exécution de la séquence d'Actes prévus dans le réseau, cela signifie que la stratégie s'avère inefficace, inappropriée ou incomplète. De cette absence découle un état émotionnel d'insatisfaction chez le tuteur. Si cet état atteint un seuil prédéfini, le tuteur doit alors amorcer une phase de délibération afin d'identifier la cause probable de l'échec, puis de modifier son comportement et mettre en œuvre une nouvelle manière de faire pour s'adapter à l'apprenant. Ainsi, le tuteur peut décider, après analyse, d'opter pour trois actions possible : il peut (1) décider de s'y prendre autrement pour tenter d'induire un autre état émotionnel, (2) décider de ne plus chercher à induire un état émotionnel particulier, car cela ne fonctionne pas, ou (3) décider de rester sur les mêmes conclusions, en signalant l'échec de l'approche.

Bien entendu, en l'absence de détecteurs biométriques ou de systèmes d'interprétation visuel du faciès de l'apprenant, la détection des manifestations émotionnelles demeure limitée. Pour ces raisons j'ai préconisé d'utiliser le Modèle OCC pour que CTS puisse se donner une approximation des états émotionnels de l'apprenant. Ces conclusions serviront d'entrées au Modèle de l'apprenant qui détient des informations plus complètes sur l'apprenant (son profil psychologique, l'historique de ses émotions, ses forces et faiblesses, ce qu'il aime et ce qu'il n'aime pas, les connaissances qu'il possède sur le domaine, le succès des interventions passées. Ainsi, c'est au Modèle de l'apprenant que revient la délicate

tâche de déterminer dans quel état émotionnel se trouve l'apprenant. Ce module n'étant pas encore implémenté dans CTS, il reste sujet à exploration.

6.3.2 Excitateurs internes

Les excitateurs internes trouvent leur source chez les modules de CTS, principalement dans le MA. Au fur et à mesure de l'avancement de la séance d'apprentissage, le MA a comme responsabilité première de faire connaître ses conclusions sur l'état émotionnel de l'utilisateur ainsi que sur son état cognitif, sur les causes probables de ces états et sur les tendances vers un état émotionnel donné. Ainsi, les excitateurs internes proviendraient d'un traitement statistique, d'une analyse causale (diagnostic) et de Perceptions immédiates de l'état émotionnel/cognitif de l'apprenant. Le fait que le Modèle de l'apprenant crée et dépose en continu dans la MT des microprocessus d'information est un processus indispensable. Il cherche à informer autant que possible le Réseau des Actes de ses constats et conclusions. La certitude de l'information, son importance ou la répétition d'un constat établissent l'urgence d'exécuter les Actes correctifs. Les émotions de l'apprenant se répercutent ainsi sur le Réseau des Actes et par conséquent sur les décisions du tuteur.

Pour illustrer ce fonctionnement, supposons que l'apprenant a connu plusieurs revers et difficultés (selon les observations d'erreurs récentes accumulées par le MA), que son comportement révèle un état émotionnel plutôt inopportun à l'apprentissage (selon l'évaluation effectuée par le Modèle OCC du MA) et que cet état d'esprit fait obstacle à son apprentissage (selon des observations historiques et selon la tendance déterminée par le MA17). Tous ces microprocessus d'informations soumis en MT résultent d'analyses effectuées par les modules de CTS et forment des excitateurs internes. Ils pourront amener le tuteur à décider de fournir un soutien émotionnel. Pour ce faire, le RA peut disposer de plusieurs sous-séquences d'Actes. Le fait que le RA soit à l'écoute des différentes

¹⁷ Le MA analyse ainsi l'apprenant de plusieurs manières et sous plusieurs angles, et non pas uniquement par le travail du modèle OCC

informations a pour conséquence d'activer variablement diverses portions du réseau. Ainsi, certaines sous séquences se retrouvent plus activées que d'autres, selon l'état de stimulation des diverses Émotions et des divers nœuds d'État dans le réseau. Si le Réseau "conclut" qu'il faut apporter davantage de soutien affectif à l'apprenant, il optera tout naturellement pour l'option la plus favorisée (la sous-séquence la plus activée) dont les préconditions sont toutes satisfaites. Les observations sur les court et moyen termes et les évaluations statistiques du MA auront ainsi généré des excitateurs internes valables pour soutenir l'apprenant. Ainsi, le traitement émotionnel que je propose pour CTS vise autant l'induction d'émotions favorables chez l'apprenant que l'exécution d'Actes pédagogiques appropriés à l'état émotionnel de celui-ci.

6.4 Description fonctionnelle

Dans cette section, j'analyse le mode de fonctionnement du système ainsi que l'interaction de certains modules qui composent CTS.

Les informations, sous forme de microprocessus d'information, transitent par la Mémoire de travail, forment éventuellement des coalitions. Dès l'instant où une information devient « consciente », elle se voit diffusée (publiée) aux modules et microprocessus de raisonnement formant les ressources de CTS.

Par exemple au fur et à mesure des actions de l'apprenant, le Modèle de l'apprenant prend connaissance des informations publiées. Le Modèle de l'apprenant à travers l'étude des actions passées de l'utilisateur, détermine l'état émotionnel/cognitif actuel de l'apprenant et traduit sa conclusion en une coalition de microprocessus d'information qu'il dépose dans la Mémoire de travail. Ceux-ci renferment les informations relatives à l'état émotionnel/cognitif ainsi qu'à l'intensité de celui-ci et forment ainsi des excitateurs internes. Dans le cas où elle est sélectionnée pour devenir consciente, cette coalition d'informations est publiée à tous les modules qui composent notre système.

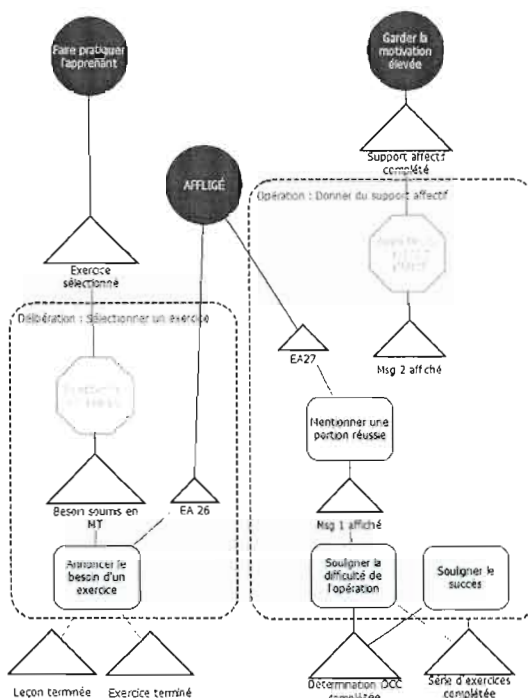


Figure 6.2 : Pouvoir discriminant des émotions. L'Émotion «AFFLIGÉ» se rattache également aux séquences «Délégation : Sélectionner un exercice» et «Opération : Donner du support affectif». De ce fait, l'état émotionnel du tuteur ne permet pas ici de choisir entre deux types d'Actes. Cependant, advenant que le contexte (les États en précondition) ainsi que le niveau d'activation du Désir «Garder la motivation élevée» favorisent la séquence de droite, l'Émotion agira alors en tant que pouvoir discriminant : les deux sous-séquences présentent les mêmes préconditions de déclenchement, mais seule celle de gauche se voit favorisée par l'Émotion «AFFLIGÉ.»

Tous les Désirs du Réseau des Actes qui reconnaissent l'information publiée vont être excités, vont élever leur niveau d'activation puis déverser différemment de l'énergie dans le réseau dans les nœuds de But qui leur sont rattachés (du haut vers le bas). L'énergie se transmet de nœud en nœud et s'accumule progressivement dans ceux-ci. Il faut voir l'énergie comme un critère de discrimination qui favorise l'exécution d'une séquence ou d'un Acte spécifique par rapport à d'autres. Ainsi, plus un nœud d'acte a accumulé de l'énergie, plus il a de chances d'être sélectionné pour exécution par rapport aux Actes

d'autres séquences. Au fur et à mesure des publications, plusieurs Émotions dans le Réseau des Actes vont s'exciter différemment, ce qui a pour conséquence d'activer différemment les séquences. Ce principe permet de prioriser certains Actes par rapport à d'autres. Or, dès l'instant où le tuteur (plus spécifiquement, le RA) "décide" comme prochaine action pédagogique d'apporter du support émotionnel à l'apprenant, il doit choisir entre plusieurs approches qui aboutissent à la même finalité (par exemple celle de donner du support émotionnel; voir **Figure 6.2** : Pouvoir discriminant des Émotions). Dans notre exemple, lors de la publication de l'excitateur interne qu'est le microprocessus d'information «Affligé», tous les liens issus de l'Émotion concernée transmettront de l'énergie, favorisant en conséquence l'exécution des sous-séquences auxquelles ils aboutissent.

Dans le cas où le tuteur décide de soutenir l'apprenant et de tenter de lui insuffler des émotions plutôt favorables, l'agent CTS s'attend à ce que son action ait un impact positif sur l'apprenant. Il revient au Modèle de l'apprenant de reconnaître les signes concernant les émotions de l'apprenant, de les interpréter et d'informer de la concrétisation ou non de l'émotion attendue chez l'apprenant. Dans le cas où l'apprenant paraît plus disposé, avec une meilleure disposition à l'apprentissage, l'agent conforte sa décision et se considère comme satisfait. Tout naturellement, les émotions concernées vont rapidement perdre leur activation. Par contre, si l'effet attendu ne se produit pas, tel que souligné par le microprocessus de confirmation qui espère l'événement, l'agent CTS augmente son niveau d'insatisfaction vis-à-vis de cette action.

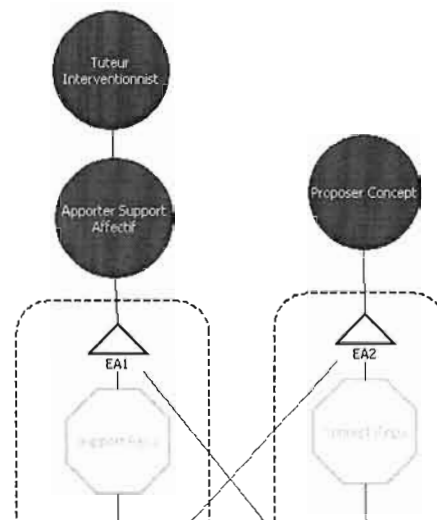


Figure 6.3 : Pouvoir discriminant du réseau

Évidemment, d'autres facteurs de discrimination dans le choix de l'Acte pourraient aussi entrer en jeu. Le Réseau des Actes présente la particularité d'être très flexible et permet d'intégrer divers aspects en agissant sur l'énergie qui parcourt le réseau. Il est ainsi possible de moduler la personnalité du tuteur (voir Figure 6.3 : **Pouvoir discriminant du réseau**).

6.5 Délibération et remise en cause

CTS ne devrait pas se limiter seulement au choix de la meilleure action pédagogique/didactique compte tenu de l'état émotionnel de l'apprenant. J'ai cherché à profiter des possibilités qu'offre l'architecture de CTS, telle la délibération. Je définis la délibération comme la capacité d'un système à poursuivre explicitement un raisonnement itératif. Elle octroie à notre agent, entre autres, la capacité de rectifier ses plans et de réajuster une approche ne produisant pas les résultats attendus. Cette capacité correspond aux conclusions de la théorie de Baars en l'occurrence.

En effet, les actions prises par le tuteur peuvent être sujettes à rectification, car elles proviennent d'inférences et de conclusions probabilistes. Ces évaluations peuvent être inefficaces, ce qui nécessite de délibérer pour trouver une solution alternative plus prometteuse. Ainsi, je confère au tuteur une capacité de remise en question quant à la pertinence de ses propres actions et de proposer, le cas échéant, une autre manière de faire apprendre.

Les excitateurs externes prennent ainsi toute leur importance. Ils vont permettre de comparer les attentes aux constats, en comparant les excitateurs externes aux excitateurs internes. Dans le cas où le système apporte du support émotionnel à l'apprenant, le système s'attend à ce qu'il y ait un retour d'information de la part de l'apprenant indiquant que celui-ci se trouve présentement dans une meilleure disposition. De même pour les choix pédagogiques, le tuteur s'attend à ce que l'apprenant démontre des compétences cognitives meilleures. Je compte utiliser des microprocessus de confirmation pour surveiller l'apparition en MT de certaines informations spécifiques confirmant ou non l'effet souhaité. Dans le cas où l'excitateur attendu ne se présente pas, un message à cet effet est émis en MT par le microprocessus de confirmation concerné. La publication éventuelle du message se traduit initialement par l'augmentation de l'activation d'une ou de plusieurs Émotions concernées dans le RA. Une répétition de l'échec produit dans CTS un état d'insatisfaction grandissant. Je confie à des microprocessus d'attention le rôle de superviser les performances de l'agent, et de permettre à ce dernier d'établir son niveau de satisfaction par rapport à ses processus et actions. Ils leur revient de solliciter le déclenchement des mesures appropriées lorsqu'ils le jugent nécessaire. Arrivé à un certain seuil, le microprocessus de raisonnement concerné n'a d'autre recours que de faire démarrer une réflexion plus profonde, une délibération.

Une délibération est un processus qui consiste à examiner les différents aspects d'un problème en vue de prendre une décision. Tous les modules composant CTS (le Réseau des Actes, le Modèle de l'apprenant, le Module du domaine, Réseau des Actes) sont appelés à participer à la délibération et à émettre leurs propositions ou leur opinion en ce qui

concerne l'état d'insatisfaction (ou tout autre objet visé par la délibération). Concrètement, pendant un nombre de cycles prédéfinis, une coalition décrivant l'état d'insatisfaction où le problème a solutionner va effectuer une boucle allant de la MT à l'ensemble des ressources par le jeu des publications de la « Conscience d'accès », et revenant en MT. Itérativement, les modules enrichissent ou s'opposent à certaines informations de la coalition, et ce jusqu'à établissement d'un consensus entre les divers modules.

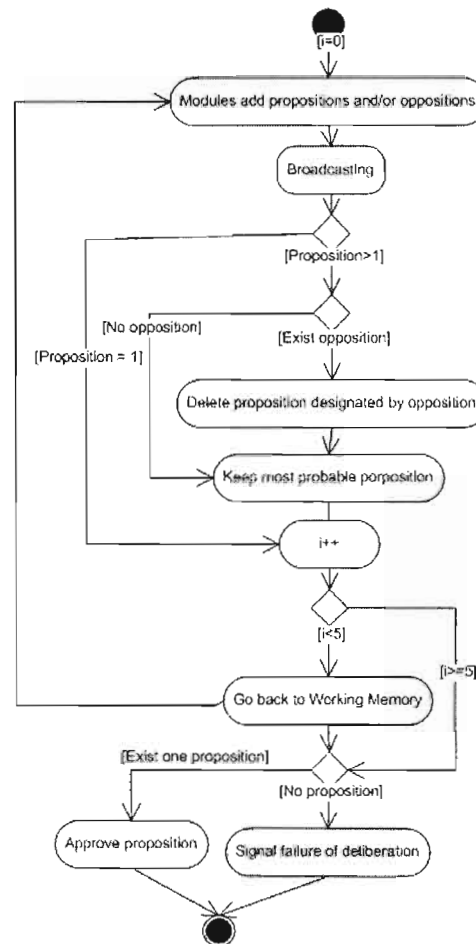


Figure 6.4 : Phase d'évolution d'une délibération

Au final, il peut arriver qu'aucune solution ne soit trouvée. Dans ce cas, le système signale l'échec de la délibération et l'action qui la sous-tend n'est plus exécutée. Toutefois, si la délibération fournit une autre alternative possible à l'insatisfaction, cela se traduit par la prise d'autres actions dans le Réseau des Actes en vue de mieux supporter l'apprenant.

6.6 Constats et tendances

Parce que les émotions sont une construction sociale qui se forme au regard des jeux des acteurs et de leurs interactions circonstanciées et quotidiennes, tout apprentissage efficient doit les prendre en considération. On s'attend d'un tuteur expérimenté qu'il conçoive une stratégie d'apprentissage propre aux conditions émotionnelles contingentes exprimées par son apprenant, et qu'il ajuste continuellement son approche, ses processus, et l'ensemble de ses comportements au regard du mouvement émotionnel ambiant. En interagissant ainsi avec l'apprenant, le tuteur cherche à faire de l'apprentissage une action de proximité, une coaction qui produit efficacement la construction du savoir. Il doit ainsi s'efforcer continuellement d'obtenir une meilleure adéquation entre contenu, méthode pédagogique et un apprenant particulier.

Pour atteindre une telle dynamique d'apprentissage, j'ai cherché à concevoir un système qui, en prenant en charge les multiples manifestations émotionnelles de l'apprenant, sait réajuster conséquemment son approche et mieux finaliser son action. L'extension que j'ai proposée à CTS, en dépit de son incomplétude, présente l'intérêt d'intégrer les aspects émotionnels dans une architecture cognitive destinée à l'e-formation.

CONCLUSION

Tout au long de mon travail de maîtrise, j'ai exploré des voies nouvelles pour l'implémentation d'un agent intelligent cognitif doté d'une conscience artificielle. C'est ainsi que j'ai contribué à l'implémentation des concepts et mécanismes fondamentaux reliés à la conscience : le Réseau des Actes, la gestion des émotions, la représentation de l'information etc. J'ai aussi eu l'occasion d'analyser de nombreux travaux relatifs au cerveau humain, ce qui m'a permis de mieux comprendre ses spécificités et ses modes de fonctionnement.

Plus spécifiquement, J'ai pu étendre l'architecture de CTS, un agent cognitif conscient et dédié au tutorat. Cette extension était loin d'être facile à entreprendre. Il s'agissait entre autres d'homogénéiser et rendre opérationnelles les différentes technologies qu'utilise CTS. Pour ce faire, j'ai apporté de nouvelles solutions pour chacune des faiblesses majeures détectées dans l'architecture initiale. J'ai été ainsi amené à développer une interface de communication bidirectionnelle pour CTS (avec l'utilisateur et avec RomanTutor), à modifier le fonctionnement du Réseau des Actes, à enrichir les actions possibles du tuteur, à simuler et à intégrer les Modèles du domaine et de l'apprenant, à rajouter des microprocessus de raisonnement spécifiques et génériques pour surveiller la Mémoire de travail, etc. Une contrainte importante lors de l'implémentation était de respecter au mieux les différents préceptes de Baars et sa description du fonctionnement de l'esprit humain.

Cependant, même si ce travail a permis de rendre opérationnelle l'agent, il n'en demeure pas moins qu'il y a encore place à l'amélioration. En raison de certaines contraintes temporelles, certains aspects du système ont été favorisés par rapport à d'autres. Ainsi, réaliser un tableau de bord pour surveiller le fonctionnement interne de CTS, implémenter un Modèle de l'apprenant apte à surveiller la progression de l'apprenant, enrichir de Réseau

des Acte afin d'offrir une plus grande richesse d'enseignement, faire une évaluation à grande échelle de CTS sur ces aptitudes à enseigner, enrichir la communication/interaction entre l'apprenant et CTS..., sont des extensions possibles pour rendre le système plus efficient.

Au-delà de ces incomplétudes et limites, je pense que l'agent CTS est actuellement apte à mener une séance d'apprentissage de qualité. En outre, comme CTS est un système extensible (*scalable*), fortement réutilisable et très polyvalent, il constitue, à mes yeux, une manière efficace de réaliser rapidement un STI cognitif performant et de bonne qualité.

D'après moi, CTS n'a pas seulement pour mission de réaliser un système tuteur intelligent. En effet, il s'agit avant tout d'une architecture d'agent cognitif qui peut donc être adaptée dans tout autre contexte de résolution de problème. Au GDAC, nous l'avons appliqué au tutorat.

Dans mes recherches futures, je m'attellerai à enrichir davantage les multiples dimensions et aspects du système afin qu'il devienne capable d'apprendre par lui-même et de s'auto corriger. Ainsi, par des extensions appropriées, le système serait à même d'élargir ses possibilités, à parfaire ses modes opératoires et à élargir son champ d'action.

BIBLIOGRAPHIE

1. Adam Carole et Evrard Fabrice. 2005. «Vers un modèle optimal des émotions : revue des modèles existants". Rapport de recherche, Vers un modèle optimal des émotions. *IRIT/2005*, juin 2005.
2. Ahn Hyungil et Picard W. Rosalind. 2005. "Affective-Cognitive Learning and Decision Making: A Motivational Reward Framework For Affective Agents". IN the *1st International Conference on Affective Computing and Intelligent Interaction*. (Beijing, Octobre 22–24). p. 866-873.
3. Alevin Vincent, McLaren M. McLaren, Sewall Jonathan et Koedinger R. Kenneth. 2006. "The Cognitive Tutor Authoring Tools (CTAT) : Preliminary Evaluation of Efficiency Gains". In *the eighth International Conference on Intelligent Tutoring Systems* (Brazil, 26-30 Juin 2006), p. 61-70.
4. Alvarado Nancy, Adams S. Samuel et Burbeck Steve. 2002. "The Role Of Emotion In An Architecture Of Mind". IBM Research.
5. Anderson R. John et Schunn D. Christian 2000. "Implication of the ACT-R Learning Theory : No Magic Bullets". In *Advances in instructional psychology*, vol. 5, p. 1-33.
6. Anderson R. John, Boyle C. Franklin et Yost Gregg. 1985. "The Geometry Tutor". In the 9th International Joint Conference on Artificial Intelligence (Los Angeles, Aout 1985), vol. 1.
7. Anderson R. John, Conrad G. Frederick et Corbett T. Albert. 1989. "Skill acquisition and the LIPS tutor". *Cognitive Science*, vol. 13, p. 467-506.
8. Anderson R. John. 1983. *The Architecture of Cognition*. Cambridge: Harvard University Press.
9. Anderson R. John. 1993. *Rules of the Mind*. Philadelphia: Lawrence Erlbaum Associates, Inc.
10. Atkinson-AbutridyA. John et Carrasco-Leon R. Julio. 1999. "An Adaptive Approach for Dynamically Generating Behavior Nets on Intelligent Agents". In *the 19th*

International Conference of the Chilean Computer Science Society. (Chilie, 1999). p. 9-18.

11. Augustin P. Jacques, Vicari M. Rosa, Pesty Sylvie et Bonneville Jean-Francois. 2004. "Applying Affective Tactics for a Better Learning". In *the European conference on artificial intelligence* (Valencia, Espagne 22-27 Août 2004). p. 109-113.
12. Baars J. Bernard 1988. *A Cognitive Theory of Consciousness*. New York: Cambridge University Press.
13. Baars J. Bernard. 1997. "In the theatre of consciousness : Global Workspace Theory of a rigorous scientific theory of consciousness". *Journal of Consciousness Studies*, vol. 4, no 4. p. 292-309.
14. Baddeley Alan, Hitch Graham. 1974. "Working Memory", *The psychology of learning and motivation: advances in research and theory*, vol. 8, p. 47-89. New York: Academic Press.
15. Bechara Antoine, Damasio Hanna, Tranel Daniel, Damasio R. Antonio. 1997. "Decision Advantageously Before Knowing of Advantageous Strategy". *Science* 28 Fevrier 1997, vol. 275, no 5304. p. 1293-1295.
16. Beck Joseph, Stern Mia et Haugsjaa Erik. 1996. "Applications of AI in education". *Crossroads, The ACM Student Magazine*, p. 11-15.
17. Belghith Khaled, Kabanza Frauduald, Hartman Leo et Nkambou Roger. 2006. "Anytime Dynamic Path Planning with Flexible Probabilistic Roadmaps". In *IEEE International Conference on Robotics and Automation* (Floride, 15-19 Mai 2006). p. 2372-2377.
18. Blackmore Susan. 2003. "Consciousness In Meme Machines". *Journal of Consciousness Studies*, vol. 10, p. 19-30.
19. Chaffar Soumaya, Chalfoun Pierre. et Frasson Claude. (2006). "La prédiction de la réaction émotionnelle dans un environnement d'évaluation en ligne". In *Technologies de l'Information et de la Communication dans l'Enseignement Supérieur et l'Entreprise* (Toulouse, 25-27 Octobre 2006).
20. Chaffar, S., & Frasson, C. (2004). Inducing Optimal Emotional state for learning in Intelligent Tutoring Systems. *Intelligent Tutoring System 2004*. Brasil.
21. Chalmers D. John. (1995). "Facing up to the problem of consciousness". *Journal of Consciousness Studies*, vol. 2, no 3, p. 200-219.

22. Crick Francis, & Koch Christof. (2003). "A framework for consciousness". *Nature Neuroscience*, vol. 6, p. 119-126.
23. Davis N. Darryl. (2004). "Why do anything ? Emotion, affect and the fitness function underlying behavior and thought". In *The Society for the Study of Artificial Intelligence and Simulation of Behavior*, (Royaume-Uni, 29-1 Mars 2004).
24. Decugis Vincent et Ferber Jacques. 1998. "Action selection in an Autonomous Agent with a Hierarchical Distributed Reactive Planning Architecture". In the *International Conference in an Autonomous Agent* (Minneapolis, Mai 1998), p. 354-361.
25. Dehaene Stanislas et Naccache Lionel. 2001. "Towards a cognitive neuroscience of consciousness: Basic evidence and a workspace framework". *Cognition*, vol. 79, p. 1 – 37.
26. Dieng Rose. 2000. "Knowledge management and the internet". *IEEE Intelligent Systems*, vol. 15, p. 14-17.
27. Dorer Klauss. 1999. Behavior Networks for Continuous Domains using Situation Dependant Motivations. In the Sixteenth International Joint Conference on Artificial Intelligence (Stockholm, Juillet 31–6 Août 1999), p. 1233-1238.
28. Dubois Daniel, Nkambou Roger et Hohmeyer Patrick. (2006). "How Consciousness Allows a Cognitive Tutoring Agent Make Good Diagnosis During Astronauts' Training". In the *Intelligent Tutoring System* (Taiwan, 26-30 Juin 2006), vol. 4053/2006, p. 154-163.
29. Dubois Daniel. 2007. "Constructing An Agent Equipped With An Artificial Consciousness: Application To An Intelligent Tutoring System". Thèse de Doctorat, Montréal, Université du Québec À Montréal.
30. Elliott Clark, Rickel Jeff et Lester James. 1999. "Lifelike Pedagogical Agents and Affective Computing: An Exploratory Synthesis." *Artificial Intelligence Today*, vol. 1600, p. 195-211.
31. Faghihi Usef, Dubois Daniel et Nkambou Roger. 2007. "Learning Mechanisms for a Tutoring Cognitive Agent". IN the 7th *IEEE International Conference Advanced Learning Technologies*, (Japon, 18-20 juillet, 2007). p. 546-547.
32. Ferrero R. Trenton. 1995. "The influence of affective processing in education and training". *New Directions for Adult and Continuing Education*, vol. 59, p. 25 - 33.
33. Fournier-Viger Philipe, Nkambou Roger et Mayers André. 2007. "Evaluating Spatial Knowledge through Problem-Solving in Virtual Learning Environments". In *Second*

European Conference on Technology Enhanced Learning (Crète, 17-20 septembre), p. 17-20.

34. Fournier-Viger Philippe, Najjar Mehdi, Mayers André et Nkambou Roger. 2006. "A Cognitive and Logic Based Model for Building Glass-Box Learning Objects". *Interdisciplinary Journal of Knowledge and Learning Objects*, vol. 2, p. 77-94.
35. Franklin Stan et Dan Jones. 2004. "A Triage Information Agent (TIA) base on the IDA Technology". In *AAAI Symposium on Dialogue Systems for Health Communication*, (Washington, 22-24 Octobre 2004).
36. Franklin Stan et Patterson, F. Gerald Jr. (2006). "The Lida Architecture : Adding New Modes of Learning to an Intelligent, Autonomous, Software Agent". In *Integrated Design and Process Technology*, (San Diego, 25-23 Juin 2006).
37. Franklin Stan, Kelemen Arpad et McCauleyLee. 1998. "IDA : A Cognitive Agent Architecture". *Systems, Man and Cybernetics*. IEEE PRESS.
38. Franklin Stan. 1995. *Artificial Minds*. Cambridge: MIT Press. 449p.
39. Franklin Stan. 2000. "Deliberation and Voluntary Action in Conscious software Agents". *Neural Network World*, vol. 10, p. 505-521.
40. Franklin Stan. 2003. "A computer Based Model of Crick and Koch's Framework for Consiousness". *Science and Consiousness Review*. no 1, p. 1-8.
41. Gaha Mohamed, Dubois Daniel et Nkambou Roger. 2008. "Adaptation d'un STI conscient par un traitement cognitif et affectif". *Revue Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, vol. 14.
42. Goleman Daniel. 1995. *Emotional Intelligence : Why it can matter more than IQ*. London, Bloomsbury Press. 343 p.
43. Hara Norika et Klang Robert 2000. "Students' distress with a Web-based distance education course: An ethnographic study of participants' experiences". *Information, Communication and Society*, vol. 3, p. 557-579.
44. Hexmoor H. Henry, Lammens M. Lammens et Shapiro C. Shapiro. 1993. "An Autonomous Agent Architecture for Integrating Unconscious and Conscious, Reasoned Behavior". In *AAAI Spring Symposium: Lessons Learned from Implemented Architectures for Physical Agents* (Californie 23–25 Mars, 1993), p. 328-336.
45. Hexmoor H. Henry. 1995. "Representing and Learning Routine activities". Thèse de doctorat à Buffalo, *University of New York*. 142 p.

46. Homeyer Patrick. 2006. "Développement d'une architecture d'agent conscient pour un système tutoriel intelligent". Thèse de Maitrise, Montréal: Université Quebec à Montréal. 107 p.
47. Jackson V. John. 1987. "Idea For A Mind". In *ACM Special Interest Group on Artificial Intelligence newsletter*. no 181, p. 23-26.
48. Johnson W. Lewis, Vilhjalmsón Hannes et Samtani Prasan. 2006. "The Tactical Language Training System". *The First Conference on Artificial Intelligence and Interactive Digital Entertainment* (Marina Del Rey, 1–3 Juin 2006).
49. Johnson W. Lewis. 2007. Extrait le 2 Janvier 2007, à partie du site web Tactical Language Trining System: www.tacticallanguage.com
50. Kabanza Frauduald, Nkambou Roger et Belghith Khaled. 2005. "Path-Planning for Autonomous Training on Robot Manipulators in Space". In the *International Joint Conference on Artificial Intelligence*, (Edinburgh, 30 Juin- 05 Juillet 2005), p. 1729-1731.
51. Kirilyuk Andrei. 2003. "Emerging consciousness as a result of complex-dynamical interaction process". *Présenté au workshop machine consciousness : complexity aspect* (Turin, 29 septembre-1octobre 2003).
52. Koch Christof. 2004. *The Quest for Consciousness. Colorado: Roberts and company publisher*. 464p.
53. Ledoux Joseph. 1996. *The emotion brain. New York: Simon & Schuster*.
54. Lester C. James, Towns G. Stuart, FitzGerald J. Patrick. 1999. "Achieving affective impact: Visual emotive communication in lifelike pedagogical agents". *International Journal of AI in Education*. p. 278 - 291.
55. Lisetti L. Christine et Nasoz Fatma. 2004. "Using noninvasive wearable Computers to recognise Human Emotions from Physiological Signals". *EURASIP Journal on Applied Signal Processing*. vol. 11. p. 1672-1687.
56. Luo Shengyuan, Jiang Ping et Zhu Jin. 2004. "Software Evolution Of Robot Control Based on OSGi". In *International Conference on Robotic and Biomimetics* (Shenyang, 22-26 Aout 2004). p. 221-226.
57. Maes Patie. 1989. "How to do the right thing". *Connection Science Journal, Special Issue on Hybrid Systems*. vol. 1180.

58. Maes Patie. 1991. "A bottom-up Mechanism for Behavior Selection in an Artificial Creature". In *the First International Conference on Simulation of Adaptive Behavior*. Cambridge: MIT Press/ Bradford Books.
59. Martinez Margaret. 2001. "Key design considerations for personalized learning on the Web". *Educational Technology and Society*, vol. 4, no1, p. 26–40.
60. Matsuura Masaru et Wada Mitsuo. 2000. "Formative Behavior Network for a Biped Robot; a Control System in consideration of Motor Development". In *Robot and Human Interactive Communication*. (Osaka, 27-29 Septembre 2000). p. 101-106.
61. McLaren M. Bruce, Kenneth R. Koendinger, Schneider Mike, Harrer Andreas et Bollen Lars. 2004. "Bootstarpping novice data : Semi-Automated Tutor Authoring Using Student Log File". In International conference on Intelligent Tutoring System. (Brézil, 30 Aout – 3 septembre 2004).
62. Miller A. George. 1956. "The magical number seven, plus or minus two : Some limits on our capacity for processing information". *Psychological review*, vol. 63, p. 81-97.
63. Negatu S. Aregahegn et Franklin Stan. 1999. "Behavioral learning for adaptative software agents". *ISCA 5th International Conference* (Denver, Colorado, Juin 1999).
64. Nkambou Roger, Belghith Kaled, and Kabanza Froduald. "An Approach to Intelligent Training on a Robotic Simulator using an Innovative Path-Planner". In *the eighth International Conference on Intelligent Tutoring Systems* (Brazil, 26-30 Juin 2006).
65. Nkambou Roger, Frasson Claude et Gauthier Gilles. 2003. "CREAM-Tools : An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems". In *Authoring Tools for Advanced Technology Learning Environments : Toward cost-effective adaptative, interactive, and intelligent educational software*. Pages 93-138.
66. Ong James et Ramachandran Sowmya. 2003. "Intelligent Tutoring Systems : Using AI to Improve Training Performance and ROI". *Stottler Henke Associates*.
67. O'regan Kerry. 2003. "Emotion and E-Learning". *Journal of Asynchronous Learning Networks*, vol. 7, no 3.
68. Ortony Andrew, Clore L. Gerals et Collins Allan. 1988. The cognitive structure of emotions. *Cambridge: Cambridge University Press*. 228 p.
69. Postle Gerard. 1993. "Putting the heart back into learning". In Boud, D., Cohen, R. & Walker, D. (Eds.), *Using Experience for Learning*, Buckingham: SRHE & Open University Press. p. 33-45.

70. Ragnemalm L. Eva. 1996. "Collaborative Dialogue with a Learning Companion as a Source of Information on Student Reasoning". In *Intelligent Tutoring system* (Montréal, 12-14 Juin 1996), p. 650-658.
71. Ramamurthy Uma, Baars J. Bernard, D'Mello K. Sideny & Franklin Stan. 2006. "LIDA : A Working Model of Cognition". In the *7th International Conference on Cognitive Modeling* (Italie, 5-8 Avril 2006). p. 244-249.
72. Rhodes Bradeley. 1995. "Pronomes in Behavior Nets". *Learning and Common Sense Section Technical Report #95-01, MIT Media Laboratory*.
73. Rickel Jeff et Johnson W. Lewis. 1997. "Intelligent Tutoring in Virtual Reality: A Preliminary Report". In the *Eighth World Conference on AI in Education* (Kobe, 19 – 22 Aout 1997), p. 294-301.
74. Ritter Steven, Anderson John, Cytrynowicz Michael et Medvedeva Olga. 1998. "Authoring Content in the PAT Algebra Tutor". *Journal of Interactive Media in Education*, vol. 98.
75. Sloman Aaron et Chrisley Ron. 2003. "Virtual machines and consciousness". *Journal of Consciousness Studies*, vol. 10, no 5, p. 133-172.
76. Starkman Neal. 2007. "*e-learning going the distance*". The journal, vol. 34, no 2, Fevrier 2007.
77. Sun Ron 1997. "Learning, Action and Consiousness : A Hybrid Approch Toward Modelleing Consiousness". *Neural Networks*, vol. 10, no 7, p. 1317-1332.
78. Thomas M. Kathleen, Drevets C. Wayne, Dahl E. Ronald, Ryan D. Neal, Birmaher Boris, Eccard H. Clayton, Axelson David, Whalen J. Paul et Casey J. Berni. 2001. "Amygdala response to fearful faces in anxious and depressed children". *Anch Gen Psychiatry*, no 58, p. 1057-1063.
79. Tyrell Tobby. 1994. "An Evaluation of Maes' Bottom-up Mechanism for Behavior Selection". *Adaptive Behavior*, vol. 2, no 4, p. 307-348.
80. Urban-Lurain Mark. 1996. "Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology". Extraite en Mai 2007, de <http://www.cse.msu.edu/rgroups/cse101/ITS/ITS.htm>
81. Zinkiewicz Lucy, Hammond Nick & Trapp Annie. 2003. "Applying Psychology Disciplinary knowledge to Psychology Teaching and Learning". LTSN Psychology Report and Evaluation, no 2.